

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

**Факультет інформатики та обчислювальної техніки**

**Кафедра технічної кібернетики**

«На правах рукопису»  
УДК 004.453

До захисту допущено:

Завідувач кафедри

\_\_\_\_\_ Ігор ПАРХОМЕЙ

« \_\_\_\_ » \_\_\_\_\_ 2020 р.

**Магістерська дисертація**

**на здобуття ступеня магістра**

**за освітньо-професійною програмою «Інформаційне забезпечення  
робототехнічних систем»**

**зі спеціальності 126 «Інформаційні системи та технології»**

**на тему: «Система моделювання гнучкого виробництва»**

Виконав:

студент II курсу, групи ІК-91мп  
Артем ЩЕРБІНСЬКИЙ \_\_\_\_\_

Керівник:

доцент, к.т.н.,  
Олег ЛІСОВИЧЕНКО \_\_\_\_\_

Консультант з нормоконтролю:

доцент, к.т.н., доц.,  
Віктор ПАСЬКО \_\_\_\_\_

Рецензент:

проф. каф. ОТ, д.т.н., проф.  
Сергій СТИРЕНКО \_\_\_\_\_

Засвідчую, що у цій магістерській  
дисертації немає запозичень з праць  
інших авторів без відповідних  
посилань.

Студент \_\_\_\_\_

Київ – 2020 року

**Національний технічний університет України**  
**«Київський політехнічний інститут імені Ігоря Сікорського»**

Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

Рівень вищої освіти – другий (магістерський)

Спеціальність – 126 «Інформаційні системи та технології»

Освітньо-професійна програма «Інформаційне забезпечення робототехнічних систем»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Ігор ПАРХОМЕЙ

«\_\_» \_\_\_\_\_ 2020 р.

**ЗАВДАННЯ**  
**на магістерську дисертацію студенту**  
**Щербінському Артему Олександровичу**

1. Тема дисертації «Система моделювання гнучкого виробництва», науковий керівник дисертації Лісовиченко Олег Іванович доцент, к.т.н., затверджені наказом по університету від « 26 » жовтня 2020р. № 3132-с

2. Термін подання студентом дисертації \_\_\_\_\_ 23.11.2020 р.

3. Об'єкт дослідження – гнучкі виробничі системи.

4. Вихідні дані – відкриті дані інформаційних порталів, статті та підручники зарубіжних та вітчизняних авторів, існуючі прототипи програмних додатків.

5. Перелік завдань, які потрібно розробити – аналітичний огляд предметної області; огляд і порівняння існуючих рішень; дослідження ефективності автоматизації; розробка системи моделювання гнучкого виробництва.

6. Орієнтовний перелік графічного (ілюстративного) матеріалу – три креслення, п'ять плакатів.

7. Орієнтовний перелік публікацій – 1 публікація.

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Перевірка на співпадіння	доцент Лісовиченко О.І.		
Нормоконтроль	доцент Пасько В.П.		

9. Дата видачі завдання \_\_\_\_\_ 27.09.2019 р. \_\_\_\_\_

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Аналіз предметної області	01.09.2020 – 08.09.2020	
2	Аналіз існуючих прототипів та проблем	08.09.2020 – 15.09.2020	
3	Детальний опис предметної області	15.09.2020 – 30.09.2020	
4	Розробка інтерфейсу програмного забезпечення	30.09.2020 – 07.10.2020	
5	Розробка алгоритмів програмного забезпечення	07.10.2020 – 31.10.2020	
6	Маркетинговий аналіз стартап-проекту	31.10.2020 – 24.11.2020	
7	Попередній захист	25.11.2020	
8	Нормоконтроль	06.12.2020	
9	Перевірка на співпадіння	09.12.2020	
10	Захист	23.12.2020	

Студент

Артем Щербінський

Науковий керівник

Олег Лісовиченко

## АНОТАЦІЯ

У роботі розглянуто проблеми управління гнучкими виробничими системами та автоматизації виробництв. Описано основні особливості та проблеми існуючих програмних комплексів «Delmia» та «AnyLogic».

Розроблено програмне забезпечення, що надає необхідний функціонал для моделювання гнучкого виробництва у 3D-просторі. Користувач самостійно вводить назви операцій, обирає типи деталей та встановлює налаштування та місцезорозташування цехів у додатку. Вбудована функція запуску симуляції дозволяє користувачу переглянути процес обробки деталей, виявити недоліки та розрахувати ефективність виробництва. Програмне забезпечення гнучке та може бути адаптоване під різні виробництва.

Ключові слова: гнучкі виробничі системи, автоматизація, моделювання.

Розмір пояснювальної записки – 107 аркушів, містить 43 ілюстрації, 23 таблиці, 8 додатків.

## ABSTRACT

The problems of management of flexible production systems and automation of productions are considered in the work. The main features and problems of the existing software packages "Delmia" and "AnyLogic" are described.

Has been developed software that provides the necessary functionality for modeling flexible production in 3D space. The user enters the names of operations, selects the types of details and sets the settings and location of workshops in the application. The built-in simulation start function allows the user to view the machining process, identify deficiencies and calculate production efficiency. The software is flexible and can be adapted to different productions.

Keywords: flexible production systems, automation, modeling.

Explanatory note size – 107 pages, contains 43 illustrations, 23 tables, 8 applications.

**Пояснювальна записка  
до магістерської дисертації**

на тему: *Система моделювання гнучкого виробництва*

Київ – 2020 року

## ЗМІСТ

СПИСОК СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ .....	9
ВСТУП.....	10
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ .....	11
1.1. Об'єкт та предмет дослідження.....	11
1.2. Огляд існуючих рішень .....	12
1.2.1. Програмний комплекс DELMIA.....	12
1.2.2. Програмний комплекс Anylogic .....	14
1.4. Постановка задачі.....	16
Висновки до розділу.....	17
РОЗДІЛ 2. АНАЛІЗ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ.....	18
2.1. Актуальність впровадження автоматизації на виробництвах .....	18
2.2. Підвищення ефективності при застосуванні автоматизації.....	19
2.3. Об'єктивно-орієнтований підхід при проектуванні автоматизованої системи керування.....	22
2.4. Гнучкі виробничі системи .....	27
Висновки до розділу.....	32
РОЗДІЛ 3. РОЗРОБКА СИСТЕМИ МОДЕЛЮВАННЯ ГНУЧКОГО ВИРОБНИЦТВА .....	33
3.1. Обґрунтування вибору середовища розробки та опис архітектури .....	33
3.2. Архітектура та структура проекту.....	40
3.3. Система інформаційних потоків та взаємодій об'єктів процесу.....	78
Висновок до розділу.....	79
РОЗДІЛ 4. МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП-ПРОЕКТУ .....	81
4.1. Опис ідеї проекту.....	81
4.2. Технологічний аудит ідеї проекту .....	82
4.3. Аналіз ринкових можливостей запуску стартап-проекту .....	83
4.4. Розроблення ринкової стратегії проекту.....	89
4.5. Розроблення маркетингової програми стартап-проекту .....	91
Висновки по розділу .....	93
ВИСНОВКИ.....	94
ПЕРЕЛІК ПОСИЛАНЬ .....	96
ДОДАТКИ.....	98
ДОДАТОК А.....	99

ДОДАТОК Б.....	100
ДОДАТОК В .....	101
ДОДАТОК Г.....	102
ДОДАТОК Ґ.....	103
ДОДАТОК Д.....	104
ДОДАТОК Е .....	105
ДОДАТОК Є .....	106
ДОДАТОК Ж.....	107



## СПИСОК СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

IoT – Інтернет речей

ШІ – Штучний інтелект

ІС – Інтелектуальна система

ЕОМ – Електронно-обчислювальна машина

МН – Машинне навчання

ЕС – Експертна система

СПЗ – Система представлення знань

БЗ – База знань

ГAB – Гнучке автоматизоване виробництво

ГBC – Гнучкі виробничі системи

АТМ – Автономний транспортний модуль

ГВМ – Гнучкий виробничий модуль

ЧПК – Числове програмне керування

VR – Віртуальна реальність

СППР – Системи підтримки прийняття рішень

AR – Доповнена реальність

AL – Програмне середовище AnyLogic

DM – Програмне середовище DELMIA

## ВСТУП

В глобальній економіці, трохи більше половини діяльностей, працюючи в яких люди заробляють приблизно 16 трильйонів доларів, мають потенціал для автоматизації шляхом впровадження штучного інтелекту в виробництво. Лише 5% всіх професій підлягають повній автоматизації за допомогою технологій, а приблизно 60% всіх професій мають 30% задач, які можуть виконуватись автоматично, без використання людської діяльності. Діяльності, що найсильніше підлягають для автоматизації, найчастіше пов'язані з фізичною працею у високо структурованих середовищах, а також у середовищах, що відповідають за збір та обробку даних. Країни з високорозвиненими економіками, високим ВВП на душу населення та проблемами, наприклад такими як швидке старіння населення, розраховують на автоматизацію, що заснована на штучному інтелекті. Це призводить до збільшення продуктивності, що, в свою чергу, впливає на підвищення ВВП. Виробництво, що працює з використанням технологій на основі штучного інтелекту, є одним з найважливіших джерел продуктивності. Для прикладу, у США, діяльності що можуть бути майже повністю автоматизовані займають 51% в економіці – це потребує виплат заробітної плати у розмірі майже 2,7 трлн. доларів. Вони найпоширеніші у виробництві, роздрібній торгівлі, продовольчій сфері та житловому секторі [12].

Автоматизація на основі штучного інтелекту відкриває безліч можливостей для діяльностей, які довго вважалися приналежністю виключно до людської сфери: розпізнавання закономірностей, прийняття рішень, обробка природної мови та візуальної інформації. Дуже часто це є засобом підвищення продуктивності а реалізація вимагає зміни бізнес-процесів. Впровадження автоматизації на основі штучного інтелекту з плином часу стає все простіше завдяки доступним бібліотекам та алгоритмам, поширенню датчиків, що генерують дані і відносно недорогим обчислювальним потужностям, що можуть бути засновані на хмарних технологіях. У промисловствих системах впровадження автоматизації та застосування штучного інтелекту підтримуються збільшенням використання датчиків та пристроїв, що підключені до інтернету речей.

## РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ

### 1.1 Об'єкт та предмет дослідження

**Об'єктом дослідження** є гнучкі виробничі системи.

**Предметом дослідження** – управління гнучкими виробничими системами.

**Актуальність.** Очікується, що високоавтономні транспортні засоби становитимуть від 10% до 15% світових продажів автомобілів до 2030 року з двозначними щорічними показниками зростання до 2040 року. Моделювання виробничих систем та їх тестування в симуляції дозволить прогнозувати технічне обслуговування, що може бути розширене за допомогою штучного інтелекту, та дозволяє доволі чітко передбачити та уникнути відмови обладнання шляхом зв'язку даних, що надходять з журналів обслуговування та інтернету речей, а також зовнішніх джерел. Очікується підвищення продуктивності активів на показники до 20%, а загальні витрати на технічне обслуговування зменшуються на 10%. Машини, що працюють з урахуванням контексту підвищують продуктивність за допомогою інтелектуальної взаємодії між людиною та системою у трудомістких умовах [14]. Підвищення продуктивності на 20% можливе для певних задач навіть коли деякі функції не повністю автоматизовані, а автоматизоване тестування якості може бути організовано з підтримкою штучного інтелекту. Глобальний ринок послуг апаратних засобів та програмного забезпечення на основі автоматизації та штучного інтелекту зросте з щорічними показниками від 15% до 25% і до 2025 року може сягнути 130 мільярдів доларів. Впровадження автоматизації та штучного інтелекту підвищить зростання продуктивності роботи підприємств в усьому світі на позначки від 0,8% до 1,4% щорічно [13].

**Новизна** полягає в розробці програмного додатку, що дасть змогу користувачу повністю змоделювати структуру гнучкого виробництва, задати всі необхідні параметри та, аналізуючи отриману симуляцію, розробити план для удосконалення автоматизації та управління підприємством, що призведе до більш раціонального використання його внутрішніх ресурсів та чіткого контролю вироблення продукції. Програмний додаток гнучкий та забезпечує широку свободу дій для користувача, що

дає змогу адаптувати симуляційні об'єкти під велику кількість підприємств схожої структури.

## **1.2 Огляд існуючих рішень**

### **1.2.1 Програмний комплекс DELMIA**

Суттю модельованих процесів в моделях систем є переміщення в часі і в просторі, як правило, великої кількості об'єктів, що утворюють в сукупності «потоки».

Результатом виробничих процесів є досить велика кількість різних «продуктів», розбитих на групи або ж одержуваних в безперервному потоковому режимі.

Мета моделювання виробничих процесів, як правило, полягає в отриманні стійкої схеми, оскільки послідовність продукції, що випускається

повторюється. Важливою процедурною концепцією аналізу ефективності є визначення періоду нестійкої роботи і усунення дефектів.

Суть концепції описується формулою «Simulation + Virtual Reality». Ключовою складовою концепції цифрового виробництва є використання певного програмного забезпечення, що дозволяє технологам здійснювати свою діяльність більш ефективно.

Було розглянуто застосування моделювання на окремих ділянках підприємства за допомогою системи DELMIA (рис. 1.1). Як і інші продукти компанії Dassault Systemes, вона ґрунтується на PPR-моделі даних про виріб (продукт, процес, ресурс). Ця модель поєднує в собі модель вироба, процеси і необхідні ресурси, що дозволяє забезпечити обмін інформацією між усіма модулями системи, в тому числі і зовнішніми. Робота з моделями CATIA і звернення до баз SMARTTEAM не вимагає додаткових коштів і відбувається в рамках концепції PPR Hub. Для взаємодії з іншими системами поставляється набір конвертерів, що дозволяють транслювати моделі в формат, доступний для DELMIA.

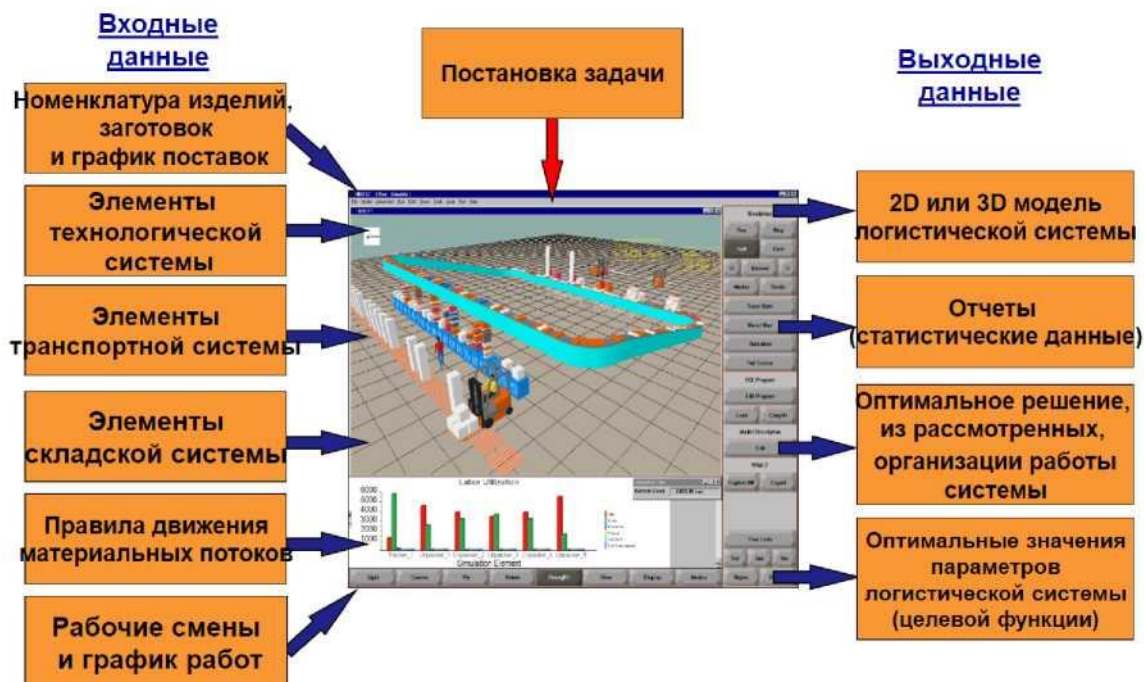


Рисунок 1.1 – Структурна схема вікна додатку імітаційного моделювання за допомогою DELMIA

Програмний комплекс DELMIA вирішує такі основні завдання:

- планування виробничих процесів;
- планування виробничих приміщень;
- аналіз ергономічності виробничих процесів;
- оптимізація виробничих процесів;
- контроль якості, нормування;
- моделювання роботи конвеєрів;
- моделювання роботи промислових роботів і роботизованих осередків;
- моделювання та верифікація технологічних процесів складання.

Було розглянуто моделювання виробничого процесу на ділянці з сучасним обладнанням в Plant Simulation, а саме, виготовлення деталей на обробному центрі MIKRON UCP 800 Duro (2 верстати на ділянці). Tecnomatix Plant Simulation - інструмент дискретного імітаційного моделювання, що дає можливість створювати цифрові моделі систем (наприклад, виробництва) для того щоб визначити характеристики системи та знайти шляхи для оптимізації її продуктивності. Створені цифрові моделі дають можливість проводити експерименти та опрацьовувати

сценарії «що якщо» без втручання в роботу існуючих виробничих систем або, коли використовуються під час планування, задовго до впровадження реальних систем. Набір інструментів для аналітики (статистичні дані, графіки та аналіз вузьких місць) допомагає оцінити різні сценарії виробництва. В результаті, отримана інформація необхідна для швидкого прийняття правильних рішень на ранніх стадіях планування виробництва.

Відома номенклатура по даному устаткуванню, річна програма випуску і час обробки деталей. Знаючи річну програму випуску кожної деталі, час на обробку, допоміжний і підготовчо-заключний час, визначається фонд часу обробки річної виробничої програми.

При симуляції об'єкти матеріального потоку отримують відповідний статус, який відображається над елементами точками. Базовий функціонал програми не дозволяє в повній мірі реалізувати моделювання виробничого процесу. Для гнучкого моделювання (передачі деталей партіями, налаштування обладнання між партіями, а також передачу деталей на термообробку) потрібно використовувати додаткові функції програми – об'єктно-орієнтоване програмування – Method [17].

### **1.2.2 Програмний комплекс Anylogic**

AnyLogic – єдина платформа для імітаційного моделювання бізнес-систем. В AnyLogic знайдуться правильні інструменти, щоб зробити модель витонченою, природною і зручною в підтримці. Візуальне drag-and-drop середовище дозволяє легко змодельовати бізнес-системи і процеси: від виробничих ліній і логістики до маркетингових компаній і соціальних змін.

Існує три основні підходи, які використовуються для створення динамічних моделей бізнес-процесів: системна динаміка, дискретно-подієвого моделювання і агентне моделювання.

Системна динаміка передбачає високий рівень абстракції і в основному використовується для задач стратегічного рівня: наприклад, щоб спрогнозувати темп

сприйняття нового товару на ринку або проаналізувати взаємозалежність соціальних процесів.

Дискретно-подієвий (процесно-орієнтований) підхід використовується в основному на операційному і тактичному рівнях, наприклад, у виробничих процесах або при оцінці інвестицій в обладнання.

Прийняття рішень, що стосуються розвитку, оптимізації або реорганізації виробництва, зумовлені багатьма факторами. Часто досить важко заздалегідь оцінити потенційний прибуток або збиток від реалізації такого роду рішень. Одним з найбільш потужних засобів для аналізу виробництва є імітаційне моделювання. Проведення імітаційних експериментів дозволяє оцінити вплив зміни різних параметрів системи і прийняти правильне рішення.

Імітаційне моделювання широко використовується у виробництві для вирішення різних проблем від оптимізації проміжних процесів до стратегічного управління. Моделювання дозволяє аналізувати не тільки конкретні процеси, а також і систему виробництва в цілому, що дає можливість перевірити капіталомісткість різних стратегій управління. Щоб не проводити експерименти в реальному житті і не заважати роботі виробництва, їх можна проводити використовуючи створену модель. (рис. 1.2).

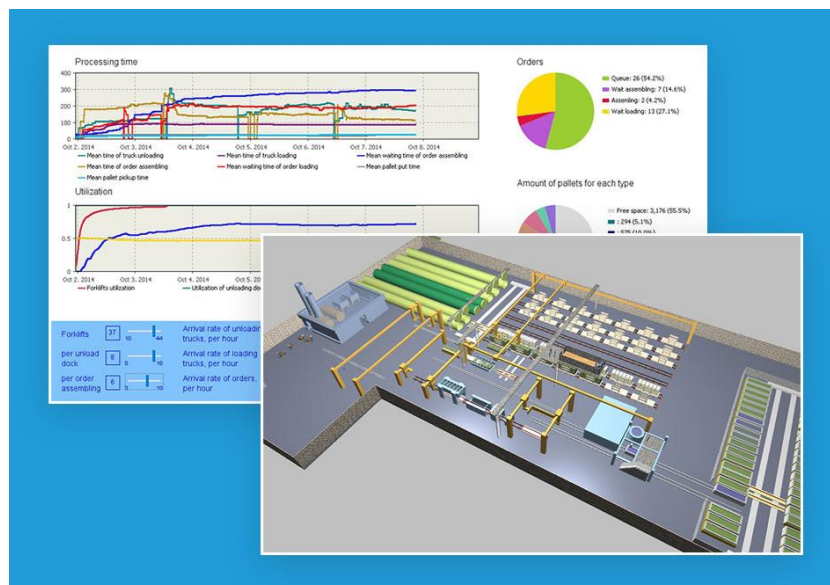


Рисунок 1.2 – Вікна статистики, анімації та візуалізації імітації роботи виробничої системи

Система AnyLogic може виконувати такі функції:

- планування виробництва;
- проектування виробничих приміщень та планування виробництва;
- вдосконалення процесів, аналіз слабких місць;
- оптимізація виробничих циклів;
- оптимізація ресурсів: персонала та обладнання;
- планування запасів (незакінчене виробництво та ресурси).

Базовий функціонал програми не дозволяє в повній мірі реалізувати моделювання виробничого процесу, роботу з центральним складом, генерацію деталей, рух вантажівок між цехами, роботму станків. Програма не володіє автоматичними алгоритмами для транспортних засобів, щоб прискорити роботу технолога, не має достатньо якісного рівня 3D моделей та не підтримує вільну «камеру спостерігача» [18].

#### **1.4 Постановка задачі**

Основними задачами магістерської дисертації є:

- Дослідження актуальності та особливостей впровадження автоматизації в управління виробничими системами;
- Аналіз ефективності автоматизації в гнучких виробничих системах;
- Дослідження підходів до проектування автоматизованих систем;
- Аналіз структури автоматизованих систем;
- Дослідження можливостей інтеграції штучного інтелекту;
- Аналіз гнучких виробничих систем;
- Розробка програмного додатку згідно нижчеперерахованим вимогам.

Графічний інтерфейс є середовищем роботи користувача з даними, тому його розробка є одним з найважливіших моментів написання якісної конкурентоспроможної системи.

Отже, програмне середовище повинно відповідати таким вимогам:

- користувач повинен бути ініціатором всіх дій;



- програма повинна швидко реагувати на команди користувача;
- додаток повинен бути інтерактивним;
- надавати образне уявлення операцій, дій;
- доступність функцій самостійного введення назв операцій;
- дружній та зрозумілий інтерфейс введення налаштувань;
- наявність функцій що упереджають випадкові дії зі сторони користувача;
- наявність функцій що дозволяють моделювати виробництво у 3D-середовищі;
- вибір об'єкта дій забезпечує доступ до засобів управління об'єктом, зміни положення об'єкта;
- наглядність подання важливої інформації при маніпулюванні об'єктами;
- однаковість методів роботи з системою;
- однаковість стилів та кольорів інтерфейсу користувача;
- наявність функцій збереження та завантаження налаштувань;
- доступність зворотньої інформації про хід процесу або режиму роботи;
- легкість освоєння і застосування;
- баланс між простотою і доступністю функцій і даних;

### **Висновки до розділу**

У розділі визначено предмет та об'єкт дослідження, детально розглянуто актуальність та описано новизну програмного продукту. Наведено цифри очікуваного зростання ефективності у виробничих галузях завдяки автоматизації та штучному інтелекту. Проаналізовано два існуючі рішення – програмі додатки Anylogic та DELMIA, описано їх основний функціонал та виявлені недоліки. На основі розглянутої інформації поставлено задачу дисертації та визначено вимоги до програмного забезпечення, що буде розроблено. Програмне забезпечення повинно враховувати недоліки розглянутих рішень.

## **РОЗДІЛ 2. АНАЛІЗ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ**

### **2.1 Актуальність впровадження автоматизації на виробництвах**

За допомогою автоматизації або автоматичного управління організації можуть виконувати процеси практично без жодної допомоги людини. Автоматизація контролює обладнання та закінчення процесів для широкого кола об'єктів та виробничих середовищ по всьому світу. Автоматизація може підвищити ефективність, якість та ефективність.

Автоматизація дозволяє перевести значну частину паперових документів в електронну форму, заощаджувати час та гроші на проведення розрахунків та ведення бухгалтерії. Крім того, автоматизація підприємства забезпечує швидкий доступ до даних по усім автоматизованим частинам виробництва та забезпечує постійний контроль показників діяльності та ефективності компанії.

Наприклад, високорозвинені економіки, з високим ВВП на душу населення, мають певні труднощі, наприклад як швидке старіння населення. Для досягнення своїх цілей по ВВП, такі економіки повинні розраховувати на автоматизацію, а на підвищення ВВП впливає збільшення продуктивності. Виробництво, яке працює на основі автоматизації та штучного інтелекту, є одним з найважливіших джерел продуктивності. Застосування автоматизації та штучного інтелекту має дуже широкий потенціал, сьогодні такі системи використовуються у багатьох сферах: промисловість, медицина, фінанси, побут та торгівля.

Автоматизовані системи управління в широко використовуються в усіх ланках управління народним господарством. Автоматизована система управління – це система «людина - машина», що забезпечує ефективне функціонування об'єкта, в якому збір та переробка інформації, що потрібна для реалізації функцій управління, здійснюється із застосуванням засобів автоматизації та обчислювальної техніки.

АСУ може вважатися тільки така автоматизована система, в якій результатом роботи є ефективне функціонування об'єкта. Об'єктом може виступати окремий технологічний процес або пристрій, сільське господарство, підприємство, галузь [5].

## 2.2 Підвищення ефективності при застосуванні автоматизації

Під ефективністю в економіці розуміється здатність до досягнення цілі з найменшими витратами. Щоб визначити ефективність, зіставляють корисний ефект та витрати на його отримання. В кожній області людської діяльності поняття ефективності конкретизується в залежності від специфічної форми шуканого ефекту. Поняття «економічний ефект» пов'язане зі змістом цілей, що переслідуються господарською діяльністю. Узагальнена мета – задоволення матеріальних та культурних потреб народу. Цій меті підпорядковуються розподіл економічних благ і виробництво. Економічний ефект виробництва вимірюється корисним ефектом вироблених благ або ступенем задоволення потреб суспільства. Показниками економічної ефективності можуть бути випуск продукції на одиницю витраченої сировини і матеріалів, продуктивність праці фондівіддача і т.п.

Ефективність АСУ визначається тим конкретним економічним ефектом виробництва, який досягається завдяки застосуванню в управлінні засобів автоматизації та обчислювальної техніки.

Високий ступінь гнучкості виробничих систем і додаткові витрати, необхідні для їх впровадження, вимагають ретельного і всебічного аналізу економічної ефективності їх використання.

Економічний ефект впровадження ГВС не завжди можна визначити простим порівнянням лише вартості та інших показників основного обладнання та агрегатів. Для підрахунку економічної ефективності впровадження ГВС не завжди підходять традиційні формули. Об'єднання в одній системі металообробки, транспортування, контролю якості та ін. не просто складає, а нелінійно збільшує економічний ефект [2].

Ефективність ГВС зростає з роками протягом певного періоду після початкових капіталовкладень. Це результат наступних факторів:

1. Набуття досвіду експлуатації ГВС;
2. Раніше впроваджені ГВС дозволять оновлювати виробництво за рахунок вдосконалення ЕОМ, програмного забезпечення і верстатів (підвищення швидкості обробки даних, збільшення обсягу пам'яті ЕОМ, розвиток мікропроцесорної техніки і т.п.);

3. Гнучкість ГВС дозволяє нарощувати виробничі потужності поступово, поетапно, обробляти одночасно кілька різних деталей;

ГВС дозволяє змінювати та удосконалювати конструкцію виробу практично без додаткових капіталовкладень. Витрати по впровадженню першої ГВС значно вище і скорочуються з впровадженням кожної наступної системи.

Повністю оцінити ефективність впровадження ГВС можливо тільки при всебічній оцінці їх технічних, організаційних, економічних переваг і соціальних наслідків.

Сучасне машинобудування приблизно на три чверті має середньосерійний та малосерійний характер виробництва. Швидко оновлюється номенклатура машин, одночасно зростає їх складність і точність; все це призводить до необхідності оперативної перебудови виробництва на підприємствах. Організаційно-технічні засоби, ефективні для масового однономенклатурного рівня виробництва, стають гальмом для відновлення продукції. Отже, необхідно створювати виробництва, що можуть швидко адаптуватися до нових умов, з високою продуктивністю праці.

Виробникам необхідний доступ до даних конкретного додатка на момент прийняття рішення. Це особливо важливо для систем моніторингу та контролю енергоспоживання, оцінки ефективності експлуатації обладнання, безпеки робочого персоналу та операторів, а також для своєчасного профілактичного обслуговування обладнання. Достовірні дані від цехів повинні надходити в режимі реального часу до потрібної людини і в потрібний час.

Виробниче середовище що розвивається вимагає здатності легко спілкуватися і користуватися вбудованими технологіями, такими як IoT і «хмари». Крім того, воно повинно володіти мобільністю і адаптувати автоматизований робочий процес до більш швидкого і інтелектуального виробничого середовища.

Зі збільшенням потоку даних і доступу до них через мережі підприємства, які стають багаторівневими, а також із зростанням використання комплексних технологій для забезпечення їх безпечної інтеграції та комплексної безпеки в точці їх використання, необхідно здійснити перехід до забезпечення безпеки, заснованої не тільки на захисті периметра підприємства.

«Розумні» пристрої, розширення використання бездротових мережевих технологій і комплексне використання даних вимагають надійної, масштабованої і стійкої виробничої структури. Оновлення мереж, про які йде мова, направлено на підтримку більш інтуїтивного апаратного і програмного забезпечення, спрямованого на підвищення автоматизації та більш ефективного доступу до даних. Такі нові мережі будуть поєднувати оптоволоконні лінії і мідні кабелі, відкриту архітектуру мережевих технологій і програмного забезпечення, що дозволяє комп'ютерній мережі, що працює в жорстких умовах навколишнього середовища, отримувати і розподіляти дані серед обладнання по всіх цехах підприємства від початкових рівнів до рівня керівництва підприємством [4].

Оновлення технології забезпечення безпеки обладнання буде пріоритетним напрямком для виробників протягом наступних кількох років. Для підвищення загальної ефективності виробництва в жорстких умовах ринкової конкуренції інтегрована в обладнання технологія забезпечення безпеки допомагає знизити частоту інцидентів, зменшити незаплановані простої обладнання, пов'язані з аваріями і помилками.

Випереджаюче управління забезпечує ефективну роботу технологічного обладнання і підтримує необхідну гнучкість і стійкість підприємства. Більш швидке реагування на вимоги споживачів в частині їх забезпечення безпечними і надійними продуктами гарантує, що виробник зможе як і раніше залишатися в своєму бізнесі. Попит на підвищення можливості оперативного контролю і якості даних зростає все швидше. Для цього потрібне поєднання гнучкої технології виробництва з одночасним управлінням витратами.

Моральне старіння обладнання і технологій є важливим питанням для виробничих структур, які вже відпрацювали великі терміни. Згідно з оцінками, 75% заводів в США мають вік понад 20 років. Щоб забезпечити наскрізну операційну ефективність, комп'ютерна апаратура і технологічне обладнання повинні бути замінені на нові, виконані на базі смарт-контролерів і драйверів.

### **2.3 Об'єктивно-орієнтований підхід при проектуванні автоматизованої системи керування**

Щороку обчислювальна техніка знаходить нові сфери застосування: сьогодні комп'ютери використовуються практично у всіх сферах людської діяльності. Постійне підвищення їх продуктивності дозволяє покладати на них рішення все більш складних завдань. Область застосування комп'ютерів вже не обмежується суто обчислювальними завданнями. Але на цьому шляху є труднощі, що виникають, зокрема, при спробах використання обчислювальної техніки областях людської діяльності, що не дуже добре нормалізуються. Основною проблемою є відсутність механізму, що дозволяє адекватно представляти як реально відбуваються процеси та їх учасників у вигляді комп'ютерної моделі. Чим точніше модель буде описувати дійсність, тим вірогідніше стануть результати, одержувані при використанні обчислювальної техніки.

Вибір системи управління бізнесом для розгляду не випадковий: сьогодні спостерігається стійкий попит на такі системи. По мірі розвитку людства форми його діяльності постійно ускладнюються. Відповідно змінюються і вимоги, що пред'являються до способів управління. Комп'ютери вже досить тривалий час використовуються в подібних системах для збору інформації, і природним є подальше розширення області їх використання для аналізу отриманої інформації та виконання базових функцій управління. Сфера управління бізнесом та проблеми що в ній виникають перестають бути прерогативою керівництва компанії і працівників. Створення подібних систем неможливо без залучення фахівців в області прикладної математики, архітекторів програмних систем.

Огляд ринку показує досить чіткий поділ споживачів в залежності від використовуваних ними систем на три категорії: персональні системи; системи для малого бізнесу; системи автоматизації діяльності середніх і великих компаній. Відповідним чином ділиться і програмне забезпечення (ПО) в цьому секторі ринку.

Системи автоматизації діяльності середніх і великих виробництв мають не тільки модулі для роботи з фінансовою інформацією, а й програми автоматизації діловодства, управління проектами, розподілу товарів по складах та інше.

Під методом проектування розуміється процес створення ряду моделей, які описують своїми засобами різні сторони розроблюваної системи. В даний час існують різні підходи до цієї проблеми, серед яких можна виділити три основні групи: метод структурного проектування; метод організації потоків даних; об'єктно-орієнтоване проектування.

На першому етапі необхідно використання методів об'єктно-орієнтованого проектування для створення об'єктів, які є відображенням реально існуючих понять предметної області і визначення базових механізмів взаємодії між ними. На другому етапі застосовуються методи структурного проектування і організації потоків даних як засобів опису взаємодії між об'єктами і документообігу на підприємствах. В силу специфіки предметної області та, як уже зазначалося, різних вимог до подібних систем, другий етап може вважатися етапом безпосереднього впровадження системи в підприємство [8].

Тому основним завданням при створенні подібних систем є задача визначення базових об'єктів і механізмів взаємодії між ними.

Найкращою системою є та, яка найбільш точно і повно описує виробничі цикли, що діють на підприємстві, і зберігає відносини між суб'єктами господарської діяльності в природній для прикладної області формі. Використання об'єктно-орієнтованого проектування при цьому призводить до необхідності виділення об'єктів, які є учасниками процесів, що відбуваються на підприємстві.

У зв'язку з довільністю вибору рівня абстракції можна буде враховувати всі об'єкти, які представляють інтерес з точки зору обліку та аналізу результатів діяльності. Об'єкти, які використовуються в системі управління бізнесом в якості представників реальних понять предметної області повинні задовольняти ряду вимог, без виконання яких процес налаштування системи буде надзвичайно складним. Розглянуті об'єкти з точки зору програмної архітектури повинні мати здатність володіти групами інших об'єктів і одночасно бути членами різних груп.

Використовуючи різні способи угруповання, користувач системи отримує можливість ведення різного виду обліку на підприємстві.

Визначення типу об'єкта не обов'язково в процесі роботи системи. Він запитується на можливість виконання якої-небудь дії. У разі підтвердження об'єкт-власник групи має право скоригувати отримані результати. При відмові можливі різні дії аж до ігнорування цього об'єкта при виконанні конкретної операції. У цьому випадку об'єкт-власник отримує можливість користуватися властивостями об'єктів групи і виконувати дії, які не були закладені в ньому на етапі створення. Наприклад, підприємство може мати склад зберігання готової продукції і цех по виробництву автомобілів. Об'єкт-завод набуває властивостей об'єкта-складу і цеху як виробничої одиниці. До нього застосовні всі операції, які можна виконувати з його складовими.

Зазвичай цей принцип використовується у множинному спадкуванні для породження об'єктів-спадкоємців, що мають властивості їхніх предків. Корінна відмінність полягає в тому, що при традиційному програмуванні це визначається на етапі створення програми. Об'єкт може виглядати як об'єкт-предок в певному контексті, але предок не існує незалежно від самого об'єкта і не може належати одночасно двом іншим об'єктам. Таким чином, необхідно, щоб для задоволення потреб користувача всі об'єкти даної системи задовольняли принципом динамічного успадкування. Принцип динамічного наслідування є ключовим фактором, який може забезпечити успіх систем такого роду.

#### Зберігання інформації.

- Об'єктно-орієнтовані БД. Одним з найбільш бурхливо розвиваючимися напрямками у галузі зберігання інформації є розробка об'єктно-орієнтованих систем управління базами даних. Вони відрізняються від реляційних БД тим, що дозволяють зберігати складні дані взаємозв'язку між їх елементами. Саме це приваблює до них інтерес як до нового способу зберігання інформації.
- Гібридні БД. Вони виступають в якості сполучного компонента між об'єктно-орієнтованими і реляційними БД, заповнюючи простір між двома технологіями. У гібридних схемах використовуються кілька підходів (від проміжного шару



програмного забезпечення до чисто реляційного підходу з трансляцією запитів, що надходять від додатків в мову SQL).

- Структуровані сховища OLE. Технологія OLE також пропонує свій підхід до зберігання інформації у вигляді об'єктів. В основу покладено принцип структурованих сховищ, що представляють собою файлову систему, побудовану поверх основної системи файлів. Вона володіє такими перевагами, як підтримка транзакцій і двійкова сумісність між платформами. Автоматично підтримується відновлення всіх файлів, які були змінені при відкаті транзакції. У разі практичної реалізації можна використовувати всі три підходи, так як операція завантаження і створення об'єктів повинна проводитися прозоро для функціональної частини системи [7].

Подання інформації.

У зв'язку з тим, що в системі з пропонованої архітектурою дані можуть зберігатися в різних форматах навіть для об'єктів одного і того ж типу, постає питання про відображення і уявленні, що зберігається. В даному випадку має сенс розмежувати розглянуті об'єкти на дві групи - об'єкти господарської діяльності як елементи структури підприємства і документи як ініціатори виконання певних дій. Аналогічно має сенс розділити і способи подання інформації – у вигляді керуючих елементів OLE (OCX) і складених документів. При цьому один вид представлення не обов'язково заперечує використання іншого способу.

Перед системою, яка повинна охоплювати всі аспекти діяльності підприємства, ставиться завдання отримання і обробки інформації, що надходить з різних джерел і має різні формати представлення. Уніфікована передача даних дозволяє не тільки обмінюватися інформацією між об'єктами OLE, а й передавати інформацію в додатки, які не підтримують цю технологію, але вміють працювати з буфером обміну даними Clipboard. Така технологія позбавляє розробника від необхідності знання того, як і звідки надходять дані. Основними методами є Query Get Data, Get Data, Set Data і Enum Format Etc. Методи Query Get Data і Enum Format Etc служать відповідно для визначення того, чи підтримує об'єкт запитуваний формат даних, і для отримання списку всіх підтримуваних об'єктом форматів.

Якщо безлічі підтримуваних форматів даних у об'єктів не перетинаються, то є можливість використання об'єктів-трансляторів. Технічно при цьому відбувається опитування реєстру операційної системи з метою пошуку об'єктів, що підтримують необхідні типи даних, і організовується послідовний процес виклику методів Get Data і Set Data. Використовуючи цей механізм, об'єкт 1 отримує можливість зберігати дані в форматі об'єкта 2, тобто в їх первинному вигляді, а обробляти в своєму власному форматі.

Таким чином, не існує обмежень на вид інформації що надходить, за умови, що в системі є відповідні транслятори в спосіб представлення, підтримуваний іншими об'єктами. Це ключ до вирішення однієї з основних проблем, яка виникає при спільному використанні різних систем, – проблеми обміну інформацією.

Говорячи про обмін даними, не можна не згадати про необхідність підтримки поновлення даних в реальному часі засобами самої системи.

Для підтримки поновлення даних доцільно використовувати метод D Advise інтерфейсу I Data Object об'єкта-сервера в сукупності з інтерфейсом I Advise Sink об'єкта-клієнта. Залежно від необхідності існує можливість встановлення одного з трьох типів зв'язку між об'єктами: «холодного»; «теплого»; «гарячого».

«Холодний» зв'язок. Такі зв'язки можуть використовуватися для обміну інформацією по заздалегідь спланованим схемам. Використання тільки методів Get Data при обміні інформацією між об'єктами може служити прикладом цього типу зв'язку.

«Теплий» зв'язок. Даний тип зв'язку між об'єктами може використовуватися, якщо для об'єкта важливий сам факт зміни даних. У цьому випадку об'єкт-клієнт знає, що інформація, якою він володіє, застаріла і може ініціювати процес оновлення через певний проміжок часу, або надіславши запит підтвердження до оператора. При встановленні «теплого» зв'язку у об'єкта сервера викликається метод D Advise і йому передається формат представлення даних, в якому клієнт хоче отримати інформацію, спосіб зв'язку - тільки повідомлення і інтерфейс прийому даних для того, щоб можна було організувати обмін даними пізніше.

«Гарячий» зв'язок. Метод з'єднання за способом «гарячого» зв'язку передбачає повідомлення клієнта про зміни інформації шляхом надсилання йому оновленої інформації. Цей метод необхідно використовувати при змінах курсів валют та іншої інформації, характер зміни якої є критичним для бізнесу.

При «гарячому» і «теплому» зв'язку існує можливість використання разового повідомлення, коли після першого надсилання інформації зв'язок між об'єктами розривається [9].

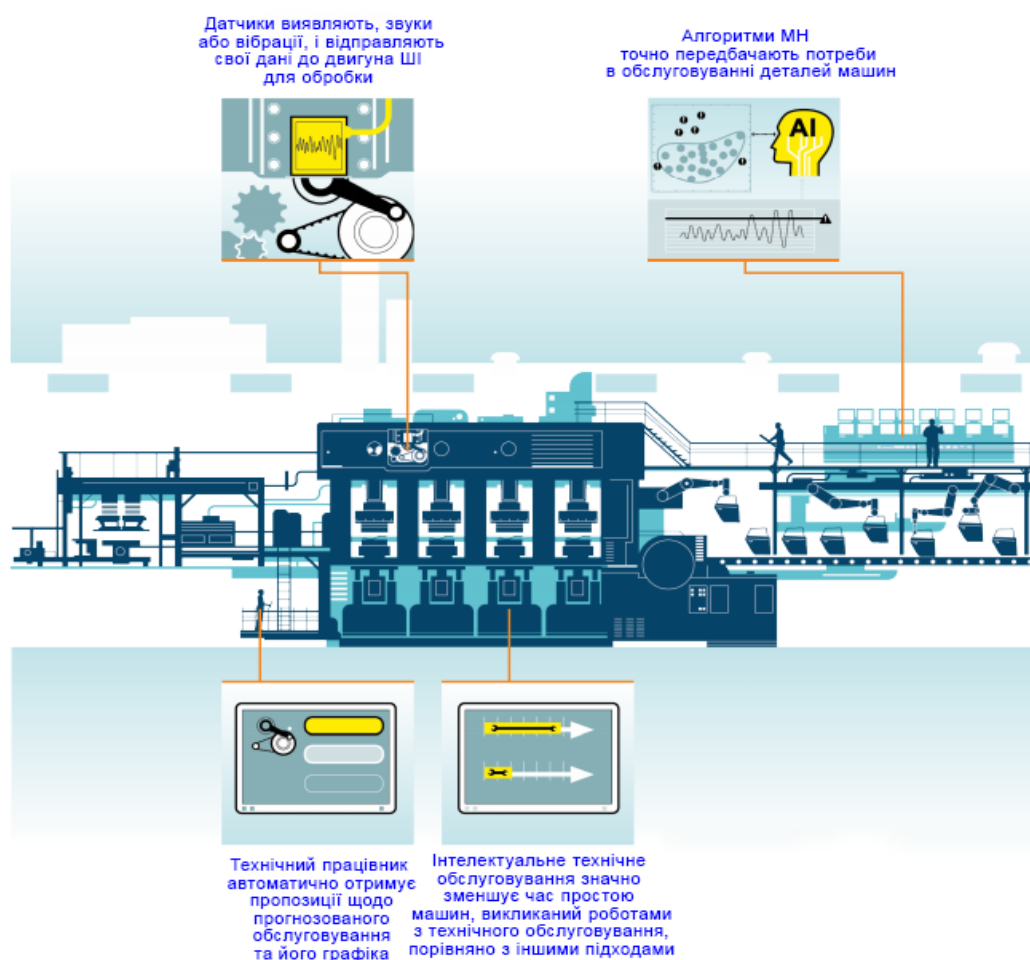


Рисунок 2.1 – схема взаємодії автоматизованої системи з оточенням та працівниками

## 2.4 Гнучкі виробничі системи

Виробничим процесом в машинобудуванні називають сукупність дій, необхідних для випуску готових виробів. Технологічний процес виготовлення

виробів покладено в основу виробничого процесу, під час якого відбувається зміна якісного стану об'єкта виробництва. Для забезпечення безперебійного виконання технологічного процесу виготовлення виробу необхідні ще й допоміжні процеси.

Основні етапи виробничого процесу:

- Отримання та складування заготовок;
- Доставка заготовок до робочих позицій;
- Різні види механічної обробки;
- Переміщення напівфабрикатів між робочими позиціями;
- Контроль якості;
- Зберігання на складах;
- Складання виробів;
- Випробування, регулювання;
- Забарвлення, обробка, упаковка і відправка.

Різні етапи виробничого процесу на підприємстві можуть виконуватися в оздоблювальних цехах або в одному цеху.

Відповідно до ГОСТ 26229 гнучка виробнича система – сукупність у різних поєднаннях обладнання з ЧПК, гнучких виробничих модулів, роботизованих технологічних комплексів, окремих одиниць технологічного обладнання та систем що в автоматичному режимі забезпечують їх функціонування протягом заданого інтервалу часу. Сукупність цих об'єктів володіє властивістю автоматизованого переналаштування при виробництві виробів довільної номенклатури у встановлених межах значень їх характеристик.

Найскладніше відбувається впровадження ГВС в складальні виробництва, це пов'язано:

- зі складністю і різноманітністю об'єктів складання і необхідної для цієї збірки оснащення;
- коротким циклом операцій складання;
- нежорсткістю або пружністю деталей;
- необхідністю в налаштуванні, підгонці та обліку малих допусків у зчленуванні деталей.

У складальних ГВС центральним компонентом є роботи з розвинуеною сенсорикою і високим рівнем машинного інтелекту, що впливає на збільшення рівня витрат при створенні збірки [1].

Вивчення сучасного виробництва, розробок і проектів показує, що спектр рішень гнучких виробничих систем простягається від виробничих модулів на базі одного верстата з ЧПУ до об'єднаних комп'ютером виробничих ділянок і цехів.

Основними класифікаційними ознаками ГАВ є:

- Масштабність структури;
- Сфера використання (за групами галузевих виробництв, видів робіт, масою і габаритами продукції);
- Технічний рівень (гнучкість, ступінь автоматизації, зростання продуктивності).

За масштабністю ГАП розділяється:

*Гнучкий виробничий модуль (ГВМ)*

Одиниця технологічного обладнання для виробництва виробів довільної номенклатури у встановлених межах значень їх характеристик з програмним керуванням, автономно функціонуюча, автоматично здійснююча всі функції, пов'язані з їх виготовленням, має можливість вбудовування в гнучку виробничу систему.

*Гнучка автоматизована лінія (ГАЛ)*

Гнучка виробнича система, в якій технологічне устаткування розташоване у відповідній послідовності технологічних операцій.

*Гнучка автоматизована ділянка (ГАД)*

Гнучка виробнича система, яка функціонує за технологічним маршрутом, в якому передбачена можливість зміни послідовності використання технологічного устаткування.

### *Гнучкий автоматизований цех (ГАЦ)*

Гнучка виробнича система, що представляє собою в різних поєднаннях сукупність гнучких автоматизованих ліній, роботизованих технологічних комплексів, гнучких автоматизованих ділянок, роботизованих технологічних ділянок для виготовлення виробів заданої номенклатури.

### *Гнучкий автоматизований завод (ГАЗ)*

Гнучка виробнича система, що представляє собою сукупність ГАЦ.

### *Роботизований технологічний комплекс (РТК)*

Сукупність одиниці технологічного обладнання, промислового робота і засобів оснащення, що функціонує автономно та здійснює багаторазові цикли. РТК, призначені для роботи в ГПС, повинні мати автоматизоване переналагодження і можливість вбудовування в систему. Засобами оснащення РТК можуть бути: пристрої накопичення, орієнтації, поштучної видачі об'єктів виробництва та інші засоби, що забезпечують функціонування РТК.

Найважливіші характеристики ГАВ:

- продуктивність;
- гнучкість;
- ефективність;

Визначаються, по-перше, характеристиками основного (верстати) і допоміжного (накопичувачі, системи автоматизованого контролю та вимірювань і т.д.) обладнання і по-друге, вдалим компонованням обладнання в системах забезпечення функціонування [3].

Продуктивність – це найважливіший показник ефективності виробничого процесу. Найбільш надійним і зручним кількісним критерієм продуктивності є продуктивність, яка вимірюється кількістю виробів, вироблених в одиницю часу (шт/год), або її зворотня величина – трудомісткість виготовлення конкретного виробу.

Прив'язка цих показників до конкретного виробу робить їх малоефективними для оцінки продуктивності процесу, з виходу якого знімаються різні вироби. ГВС виробляє не тільки різні деталі, але і різний їх число в одиницю часу.

Продуктивність можна розглядати без таких понять як гнучкість і мобільність.

Гнучкість автоматизованого виробництва це:

- можливість обробляти на одній і тій же технологічній лінії різні деталі в різних поєднаннях;
- можливість зміни в будь-який момент стратегії виробництва в залежності від необхідності;
- зміна оброблюваних деталей без залучення додаткових значних витрат;
- зміна складу технологічної лінії в залежності від вимог;
- повторне використання значного відсотка існуючих капіталовкладень в тому випадку, якщо доводиться повністю міняти тип продукції.

Гнучкість і продуктивність – це два фактори, які дуже важко об'єднувати, і тому тільки з аналізу цих факторів можна визначити їх оптимальне співвідношення для об'єднання, і цей аналіз повинен виконуватися спільно конструктором і споживачем.

Цей аналіз повинен сприяти визначенню того, як і наскільки гнучка система виробництва може впливати і скорочувати собівартість продукції, де під собівартістю продукції розуміється як пряма вартість виробництва, так і всі непрямі витрати виробництва, які можуть бути змінені завдяки застосуванню цієї нової сучасної системи виробництва.

Гнучкі виробничі системи зазвичай складаються з певної кількості верстатів, системи транспортування і розвантаження деталей і системи управління, що складається з однієї або декількох ЕОМ і відповідного математичного забезпечення [10].

Верстати можуть бути спеціалізовані або універсальні, однакові або різні, більш чи менш гнучкі, оснащені чи ні будь-якою особливою апаратурою.

Система транспортування може бути організована для транспортування деталей, оснастки, палет (супутників) або ж тільки для перевезення деталей; може

бути більш жорсткою (наприклад, лінія на роliках з приводом), або ж більш гнучкою (наприклад, самохiднi вiзки на рейках або з управлінням по дроту, може виконувати тiльки подачу окремих деталей, а потiм роботи будуть забирати цi деталі і закріплювати або знімати їх на оснащених верстатi, може виконувати перевезення тiльки деталей, або також і перевезення інструментiв.

Система управління може бути найпростішою (управління тiльки одним рухом вiзків або деталей) або може ускладнюватися і бути системою, яка управляє програмою обробки деталей, магазином з інструментами, якістю обробки, стратегією, яка змінюється в залежності від вимог виробництва, може бути складною системою комплексного управління цехом з усіма його складовими частинами [8].

### **Висновки до розділу**

У розділі розглянуто актуальність впровадження автоматизації на виробництвах. Це дає змогу забезпечити швидкий доступ до даних по усім автоматизованим частинам виробництва та постійно контролювати показники діяльності та ефективності компанії, результатом роботи якої є ефективне функціонування об'єкта.

Описано підвищення ефективності при впровадженні автоматизації у виробництво, ефективність АСУ визначається тим конкретним економічним ефектом виробництва, який досягається завдяки застосуванню в управлінні засобів автоматизації та обчислювальної техніки.

Розглянуто об'єктно-орієнтований підхід до проектування автоматизованої системи керування виробництвом, чим точніше модель буде описувати дійсність, тим вірогідніше стануть результати, одержувані при використанні обчислювальної техніки.

Проаналізовано структуру автоматизованих систем, їх взаємодію зі штучним інтелектом та структуру гнучких виробничих систем.



## **РОЗДІЛ 3. РОЗРОБКА СИСТЕМИ МОДЕЛЮВАННЯ ГНУЧКОГО ВИРОБНИЦТВА**

### **3.1 Обґрунтування вибору середовища розробки та опис архітектури**

Оскільки основною магістерської дисертації є розробка програмного забезпечення, що надасть змогу моделювати роботу гнучкого виробництва у 3D-середовищі, обирати налаштування обробки деталей та, у разі потреби, адаптувати середовище до різних систем, перед початком роботи над поставленою задачею було розглянуто та вивчено найпоширеніші середовища розробки, за допомогою яких можливо розробити програмне забезпечення з вказаною функціональністю з підтримкою 3D-простора. Для обрання найзручнішого варіанту, були поставлені наступні критерії: підтримка імпорту 3D-моделей ззовні, можливість створення 3D-моделей у середовищі, широка функціональність, дружній інтерфейс, належність навчальних функцій та підказок, оптимальна для вивчення мова програмування, підтримка функціональності штучного інтелекту, безкоштовне використання.

Проаналізувавши можливості багатьох середовищ, широту та зрозумілість документації, інструментарій, легкість розробки, підтримку функцій штучного інтелекту, належність навчальних функцій та підказок, зрозумілість інтерфейсу та доступність середовища, для розробки програмного забезпечення було обрано рушій Unreal Engine.

Unreal Engine – ігровий рушій, що розробляється та підтримується з 1998 року компанією Epic Games. Це найпопулярніше середовище для розробки додатків у 3D-просторі та ігор. Оскільки Unreal Engine зосереджує в собі надзвичайно велику аудиторію розробників, на різних платформах майже завжди знаходиться рішення будь-якої проблеми, що може виникнути в процесі розробки. Існує окремий форум, wiki, сервіс для запитів та відповідей AnswerHub та є можливість приєднатися до спільних проектів розвитку через GitHub. На відеосервісах та на інтернет-форумах завантажено багато літератури та навчальних курсів, в яких детально описані всі інструменти та аспекти роботи з середовищем. Широкий асортимент функцій для різних цілей. Написаний на мові C++, рушій дозволяє створювати додатки для

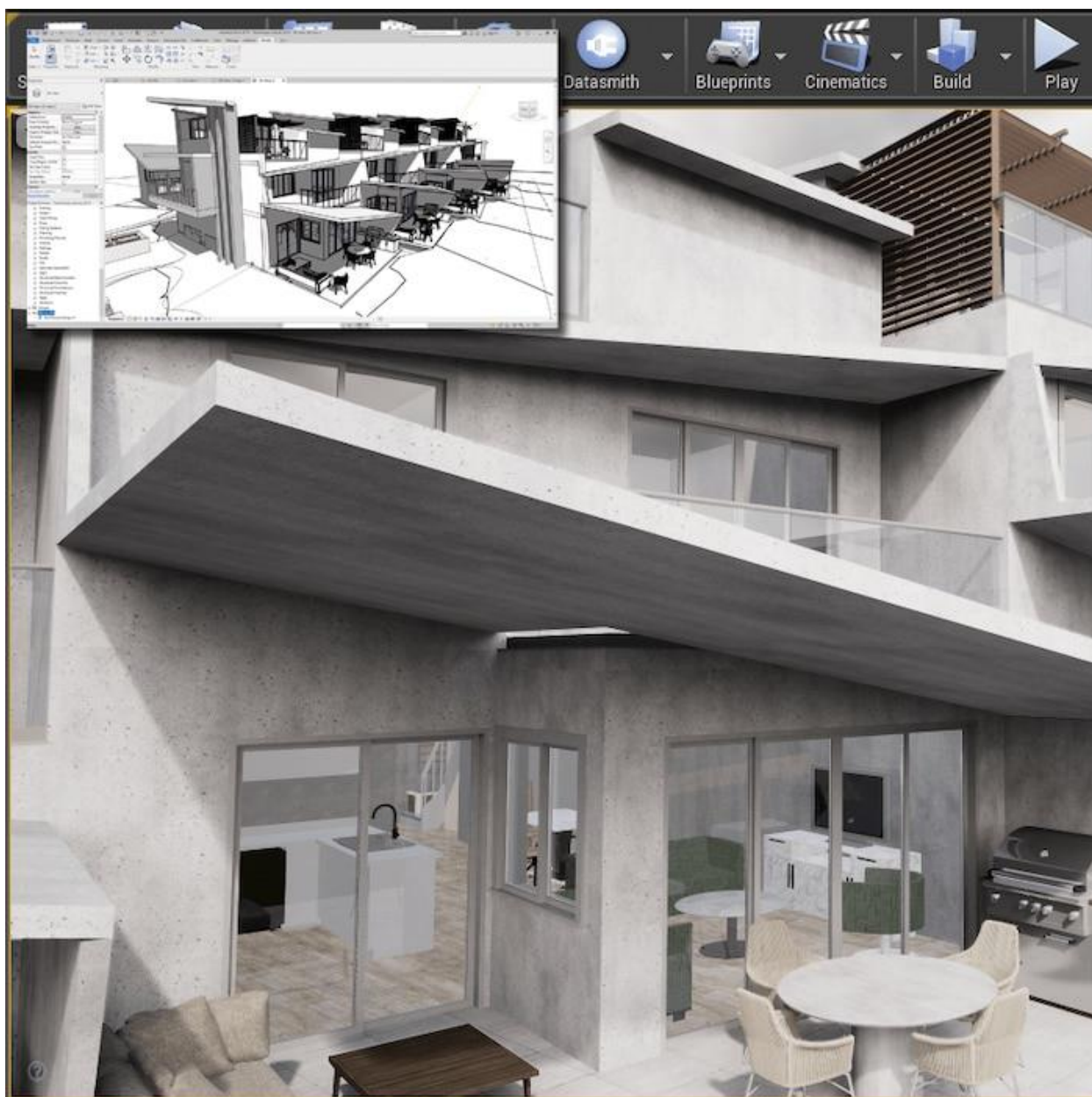
більшості операційних систем і платформ: Microsoft Windows, Mac OS і Mac OS X, Linux, консолей Xbox, PlayStation (5,4,3,2), PS Vita, PSP, Wii, GameCube Dreamcast та ін., а також на портативних пристроях, наприклад, на пристроях операційної системи Android та IOS. Середовище повністю безкоштовне на деяких умовах, а саме доти, доки додаток, зроблений за допомогою Unreal Engine не заробить 1.000.000\$. Після проходження цього порогу, компанія-розробник повинна сплачувати 5% від доходу. Така політика робить середовище дуже привабливим для розробників невеликих проєктів, учнів шкіл, студентів та інших.

Для всіх користувачів рушія відкрито Unreal Engine Marketplace. Marketplace – це цифровий магазин, який дозволяє розробникам та творцям вмісту надавати художні активи, моделі, звуки, середовища, фрагменти коду та інші функції, які інші користувачі можуть придбати разом з підручниками та іншими посібниками. Деякий контент Еріс надає безкоштовно, включаючи раніше пропоновані ресурси та підручники по Unreal (рис. 3.3).

Для спрощення портування рушія використовує систему модульно залежних компонентів, підтримує різні технології рендеринга: OpenGL, Direct3D, Pixomatic, відтворення звуку: OpenAL, EAX, DirectSound3D, засоби розпізнавання мови, голосового відтворення тексту, модулі для роботи з мережею та підтримуваних пристроїв введення.

Для зв'язку по мережі підтримуються технології Windows Live, GameSpy, Xbox Live та інші, працюючи з 64 клієнтами одночасно.

В Unreal Engine присутній повний набір засобів розробки для тих, хто працює з технологіями реального часу. Від візуалізації дизайну та кінематографічного досвіду до високоякісних додатків на ПК, консолі, мобільному пристрої, VR та AR – Unreal Engine надає все необхідне для роботи у цих напрямках. Для виробництва медіа, рушія підтримує такі галузеві стандарти як FBX, USD та Alembic. Підтримка USD дозволяє користувачам краще співпрацювати з членами команди та працювати паралельно. Unreal Engine може читати файл USD з будь-якого місця на диску без трудомісткого повного імпорту та записувати зміни до нього; миттєво оновлює зміни, внесені іншими членами команди.



*Рисунок 3.1 – Вікно Unreal Editor під час роботи з архітектурою*

За допомогою двигуна можливо конвертувати цілі сцени, включаючи анімацію та метадані, із 3ds Max, Revit, SketchUp Pro, Cinema 4D, Rhino, SolidWorks, Catia та цілого ряду інших форматів DCC, CAD та BIM з високою точністю. Unreal Engine також включає Unreal Editor, інтегроване середовище розробки, доступне в Linux, MacOS та Windows для створення вмісту. Є система підтримки оптимізації в режимі реального часу, рушій пропонує такі інструменти, як автоматичне створення LOD (рівень деталізації). Спеціальні функції, що усувають приховані поверхні, текстури та непотрібні моделі (рис 3.1).

Вбудовані редактори візуальних ефектів Cascade та Niagara дозволяють повністю налаштувати систему часток, використовувати широкий спектр модулів.

Всі елементи рушія представлені у вигляді об'єктів які мають набір характеристик та класів, які визначають доступні характеристики [15]. Будь-який клас є «дочірним» класом *object*. Основні класи та об'єкти:

Актор (*actor*) – «діючий об'єкт» або «суб'єкт», є перекладом з англ. *Actor* - той, хто діє - батьківський клас, що містить всі об'єкти, які мають відношення до процесу роботи додатку і мають просторові координати.

Пішак (*pawn*) – від англ. *pawn* – той, ким маніпулюють – фізична модель об'єкта або гравця, керованого штучним інтелектом. Метод управління описаний спеціальним об'єктом - контролером. Контролер штучного інтелекту описує лише загальну поведінку пішаків під час роботи додатку в реальному часі, а такі параметри як відстань, на якому пішак звертає увагу на інші об'єкти або кількість пошкоджень, після яких пішак перестає функціонувати, задаються для кожного об'єкта окремо.

Світ, рівень (*world, game level*) – об'єкт, що характеризує загальні властивості «простору» – фізику, силу тяжіння та атмосферні явища простору, в якому розташовуються всі «актори». Також може містити в собі параметри процесу, як, наприклад, спеціальний режим, для якого призначений рівень.

Широкий інструментарій редагування сітки та анімації дозволяє повністю налаштовувати акторів на сцені, у тому числі є системи, такі як *state machines* (пристрій, що може бути в одному із заданих станів залежно від його попереднього стану та від значень параметрів. Для роботи з простими і нерухомими об'єктами простору (наприклад, стіни) використовується двійкове розбиття простору – простір ділиться на «порожній» та «заповнений». У «порожній» частині простору розташовуються всі елементи, а також тільки там може знаходитися «точка спостереження» при відображенні сцени. Не виключається можливість часткового або повного переміщення об'єктів в «заповнену» частину простору, однак це може призвести до неправильної обробки таких елементів (наприклад, розрахунок колізій, фізичної взаємодії) або неправильного відтворення в разі переміщення туди «точки спостереження» (наприклад, ефект «залу дзеркал»). Якщо пішаки потрапляють в «заповнену» частину простору, вони відразу «гинуть».

Поверхня (surface) є основним елементом двійкового дерева простору. Ці елементи створюються на межі перетину між «порожньою» і «заповненою» частинами простору. Комплект елементів двійкового дерева простору має назву нод (node, з англ. вузол). Цей термін вживається в контексті node count – кількість нод на екрані або в просторі загалом. Кількість нод, що одночасно відображаються на екрані, впливає на продуктивність при рендері сцени. Якщо якийсь нод не потрапляє в область екрану або перекривається цілком іншими нодами, він не обраховується – це сприяє підвищенню продуктивності, особливо в закритих просторах.

Опис «порожніх» і «заповнених» частин простору виконується за допомогою комплекту замкнених тривимірних об'єктів, складених з непересічних поверхонь – Брашей (brush, з англ. пензлик). Такий принцип побудови простору має назву конструктивною суцільною геометрією. Геометрія може бути «від'ємною» (спочатку заповнений матерією простір) і «адитивною» (весь простір спочатку «порожній»).

Конструктивна блочна геометрія, КБГ (англ. Constructive Solid Geometry, CSG) – технологія, яка використовується в моделюванні твердих тіл. Конструктивна блочна геометрія часто, але не завжди, є способом моделювання в тривимірній графіці і САПР (система автоматичного проектування). Вона дозволяє створити складну сцену або об'єкт за допомогою бітових операцій для комбінування декількох інших об'єктів. Це дозволяє більш просто математично описати складні об'єкти, хоча не завжди операції проходять з використанням тільки простих тел. Так, часто за допомогою конструктивної блокової геометрії представляють моделі або поверхні, які виглядають візуально складними; насправді, вони є трохи більше ніж розумно скомбіновані або декомбіновані прості об'єкти. У деяких випадках конструктивна блокова геометрія виповнюється за допомогою полігональних сіток (англ. Polygonal mesh), і може бути процедурною або параметричною.

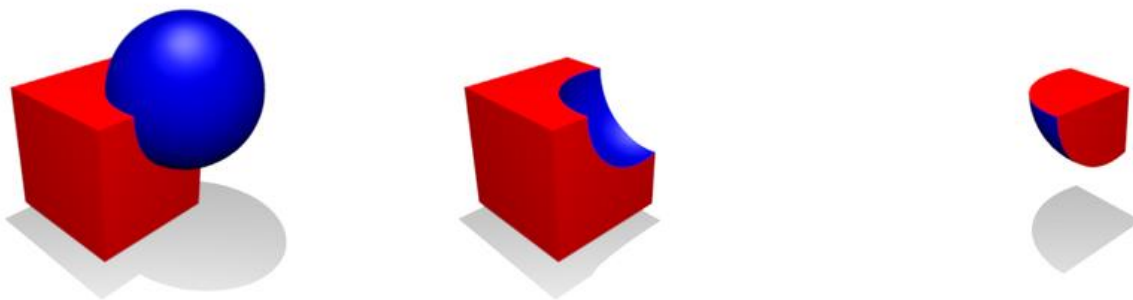
Система автоматизованого проектування – це система, що реалізує інформаційну технологію функцій проектування, представляє собою організаційно-технічну систему, що призначена для автоматизації процесу проектування та складається з персоналу і комплексу програмних, технічних та інших засобів автоматизації його діяльності.

Програмне забезпечення САПР використовує векторну графіку для механічного проектування з метою зображення об'єктів традиційного креслення. Також може створювати растрові дані, що відображають загальний вигляд об'єктів що проектуються. Але це включає в себе більше, ніж просто шаблонні форми. Так само як і при ручному створенні інженерних та технічних креслень, вихідні дані САПР повинні передавати інформацію, таку як характеристики використовуваних матеріалів, розміри і допуски, процеси, відповідно до вимог для конкретних додатків.

CAD може використовуватися для проектування фігур і кривих в двовимірному (2D) просторі; або кривих, твердих тіл і поверхонь в тривимірному (3D) просторі.

САПР є важливою складовою в промисловому конструюванні, широко використовуваною в різних галузях, в тому числі аерокосмічній, автомобільній та суднобудівній промисловості. У промисловому і архітектурному проектуванні, протезуванні та багатьох інших. Також САПР широко використовується в створенні рекламних і технічних матеріалів, в комп'ютерній анімації для спецефектів у фільмах та ін. Через свою велику економічну важливість, САПР є основною рушійною силою досліджень в області дискретної диференціальної геометрії, обчислювальної геометрії та комп'ютерної графіки (як апаратної, так і програмної).

Найпростіші тіла, що використовуються в конструктивній блоковій геометрії – примітиви (англ. Primitives), тіла з простою формою: куб, циліндр, призма, піраміда, сфера, конус. Побудова більш складного об'єкта відбувається шляхом застосування до описів об'єктів булевих (двійкових) операцій на множинах – об'єднання, перетин і різниця (рис. 3.2).

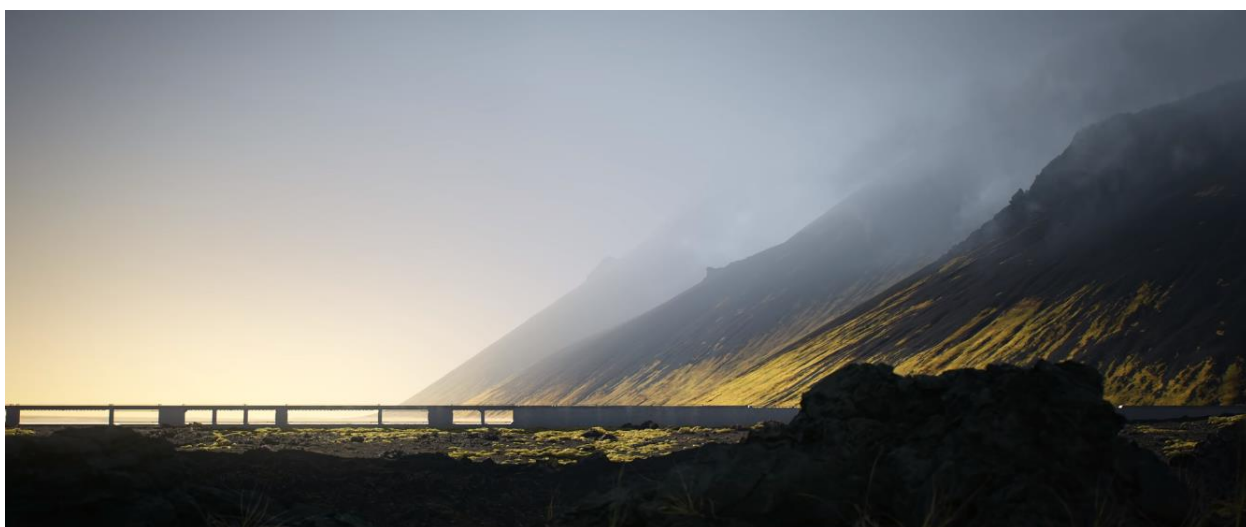


*Рисунок 3.2 – Булеві операції об'єднання, різниця та перетин відповідно*

Примітив, як правило, може бути описаний процедурою, яка приймає деякі значення параметрів, наприклад, для побудови сфери досить знати її радіус і положення центру.

Браши діляться на такі типи:

- суцільні (solid) – повноцінно беруть участь в двійковому розбитті простору.
- адитивні (additive) – «заповнюють» двійковий простір.
- від’ємні (subtractive) – «видаляють» обсяги в просторі.
- напівсуцільні (semi-solid) – впливають на фізичну модель двійкового дерева простору, але не впливають безпосередньо на двійкове дерево. Можуть тільки «заповнювати» простір. Існують для створення «невидимих» перешкод, а також зниження числа нод та полігонів.
- порожні (non-solid) – не впливають на двійкове дерево простору, тільки створюють поверхні. Використовуються в основному для створення обсягів (volume) – частин простору, які мають властивості, відмінні від властивостей всього іншого світу. Обсяги мають параметр пріоритетності: до акторів що знаходяться в обсягах з великим пріоритетом можуть застосовуватись певні властивості. Світ завжди має мінімальний пріоритет. Обсяги використовуються для створення симуляції води, туману, зміни параметрів гравітації, в'язкості і т.п.



*Рисунок 3.3 – Приклад професійного рівня візуальних ефектів в Unreal Engine*

Важливою складовою середовища є вбудований штучний інтелект. Актори, що контролюються за допомогою штучного інтелекту, мають у собі підсилену



просторову обізнаність про навколишній світ. Це дає їм можливість рухатися за допомогою системи Unreal Engine та системи штучного інтелекту, що керується за допомогою мови програмування Blueprints або за допомогою дерева поведінки. Крім того, налаштовані функції, що динамічно будують навігаційну сітку на основі отриманих даних у реальному часі [16].

### 3.2 Архітектура та структура проекту

#### Головне меню



*Рисунок 3.4 – Вікно головного меню програми*

Перше, що бачить користувач після запуску додатку – головне меню (рис. 3.4), що побудоване максимально просто та зрозуміло, та складається з 5-ти кнопок, з якими можна взаємодіяти:

- ComboBox в лівій частині меню – це опція обрання завантаження одного з файлу збережених раніше налаштувань. Інформація з ComboBox зчитується тільки у випадку коли буде натиснута кнопка «Start simulation».
- «Start simulation» – це кнопка запуску симуляції на основі збережених налаштувань з файлу з ComboBox, список цих файлів може редагувати та доповнювати сам користувач у інших пунктах меню.



- «Change settings» – кнопка переходу до системи встановлення налаштувань, яка складається з чотирьох стадій, а саме: введення назв всіх операцій для деталей; встановлення типів деталей, набору введених операцій для кожної з них, загальної кількості для обробки кожного типу; встановлення кількості цехів, кількості станків всередині кожного цеха, операцій, що повинен виконувати кожен з станків для деталей та часу, що займає кожна операція для кожного станка; система генерації та позиціювання кожного цеху у відносних координатах світу на площині.
- «Save file manage» – кнопка що відкриває додаткове вікно, в якому доступна функція видалення файлів збережених налаштувань, окрім того, який доступний по замовчуванню.
- «Quit» – кнопка виходу з програми.

### **Функціональність кнопки «Start simulation»**

Натиснувши цю кнопку, на основі отриманої інформації з ComboBox, програма одразу завантажить порожню локацію та до неї з раніше збереженого файлу налаштувань та запустить симуляцію відповідно обраному файлу (рис. 3.5).

Програма запускає 3D-симуляцію обраних налаштувань (рис. 3.13). У верхньому лівому кутку екрана користувач одразу побачить підказку, яка допоможе зрозуміти, які ключові клавіші відповідають за керування камерою спостерігача. Натиснувши клавішу «ECS», користувач побачить додаткове вікно меню, під час якого вся симуляція буде поставлена на паузу (рис. 3.6).

Клавіша «H» надає функцію приховання та відображення всіх маркерів з назвою та інформацією деталей, цехів та станків (рис 3.7).

Натиснена клавіша «TAB» відображає статус усіх згенерованих на момент натиснення деталей та їх розподілення по всіх цехах у форматі [Тип деталі] – [наступна операція або DONE]. Ключове слово «DONE» відображається замість назви наступної операції у випадку, коли деталь пройшла всі операції та готова до транспортування до головного складу або вже присутня на ньому (рис. 3.8).

Спочатку збирається інформація по кожному з цехів, що в програмі мають кодові імена «Workshop». При отриманні посилання для кожного цеха, алгоритм

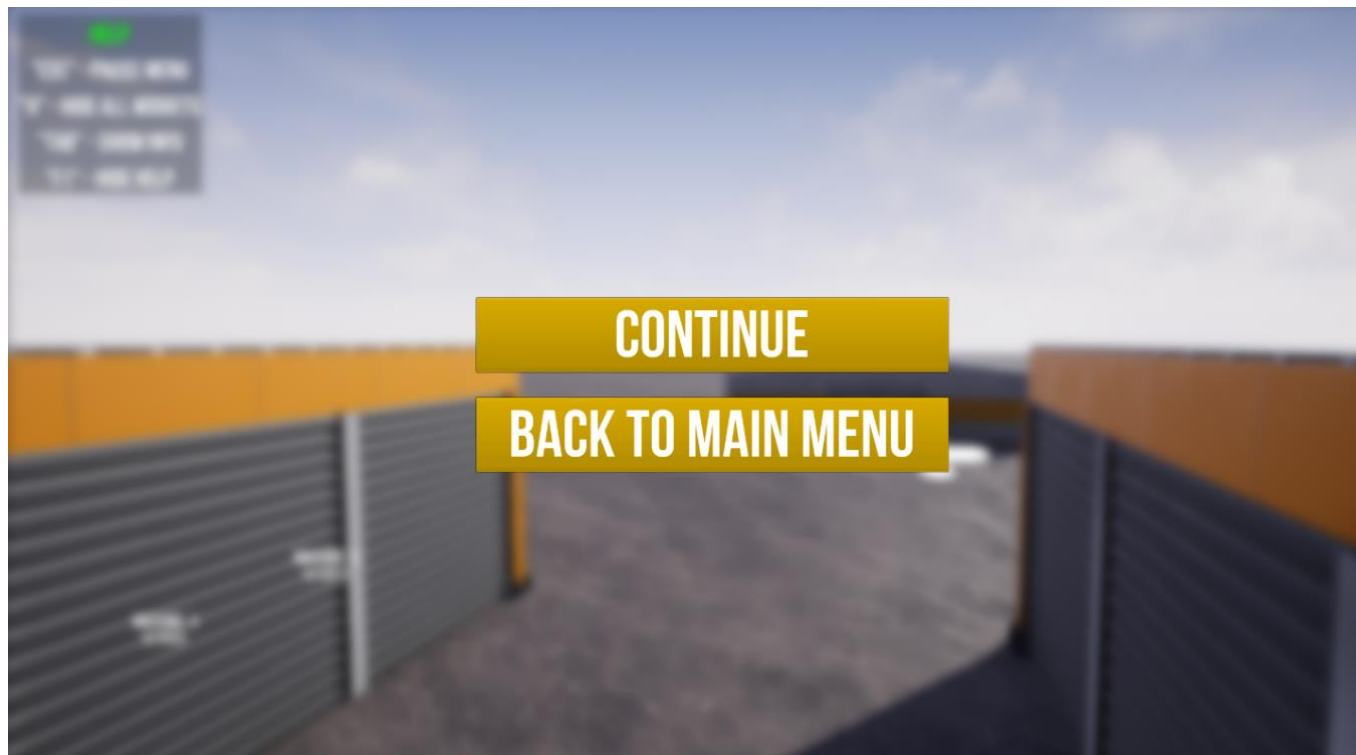
створює новий віджет, що побудований спеціально для виведення на екран цехів, звертається до спеціального об'єкта TriggerBox, що охоплює кожен цех. TriggerBox повертає інформацію про всіх акторів класу «Engine» що він перекриває. По циклу ForEach для кожної деталі, отримується назва типу деталі, наступна операція та кількість операцій що залишилась для проходження повного циклу обробки. Для кожної деталі створюється спеціальний віджет, що складається з трьох пустих полів у рядок з відступами, в які записується отримана інформація. Створені віджети поступово додаються до батьківських віджетів Workshop. Відповідно кожен з віджетів Workshop додається до основного батьківського віджету, що розгортається на весь екран. Після завершення описаних операцій, алгоритм звертається до актора класу Storage, отримує від нього посилання на об'єкт TriggerBox, що повертає посилання на всіх охоплених акторів класу «Engine». Алгоритм створює новий віджет, що побудований спеціально для виведення на екран деталей що знаходяться на складі. По циклу ForEach для кожної деталі, отримується назва типу деталі, наступна операція та кількість операцій що залишилась для проходження повного циклу обробки. Створені віджети поступово додаються до батьківського віджету Storage. Після завершення додавання, віджет Storage додається до основного батьківського віджету (рис. 3.12).

Клавіші «-» та «+» відповідають за зміну швидкості роботи усієї симуляції. Тобто, якщо користувач бажає прискорити симуляцію, щоб швидко дійти до ключового моменту, або навпаки, зробити симуляцію повільніше, щоб було більше часу роздивитися ситуацію, то за допомогою двох клавіш легко можна змінювати швидкість роботи програми. При цьому, швидкість камери спостерігача не зміниться (рис. 3.11). Індикатор зміни швидкості плину часу виводиться у верхній частині вікна програми з позначкою «SPEED» та числом, що позначає у скільки разів змінена швидкість (рис. 3.9).

Клавіша «F1» відповідає за приховання або відображення вікна підказок, що знаходиться у лівому верхньому кутку екрана (рис. 3.10).



*Рисунок 3.5 – Приклад роботи програми після запуску симуляції рівня на основі налаштувань по-замовчуванню*



*Рисунок 3.6 – Вікно паузи та додаткового меню*



*Рисунок 3.7 – Приклад відображення роботи програми після запуску симуляції та відключення маркерів натисненням на клавішу «H»*



*Рисунок 3.8 – Відображення списку деталей після натискання клавіші «TAB»*



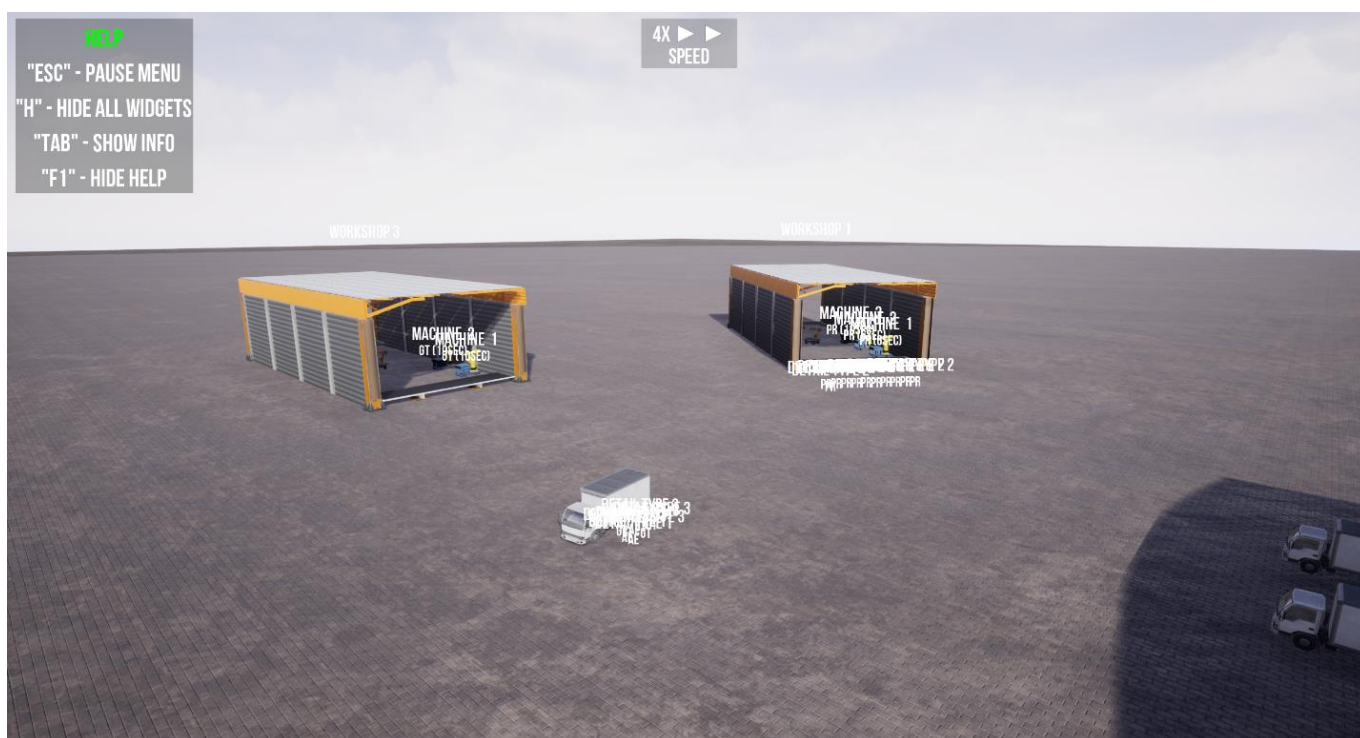


Рисунок 3.9 – Робота симуляції зі збільшеною у 4 рази швидкістю



Рисунок 3.10 – Вікно програми з прихованим віконцем підказок у лівому верхньому кутку екрана після натиснення клавіші «F1»



Рисунок 3.11 – Блок-схема алгоритму зміни швидкості плину часу симуляції

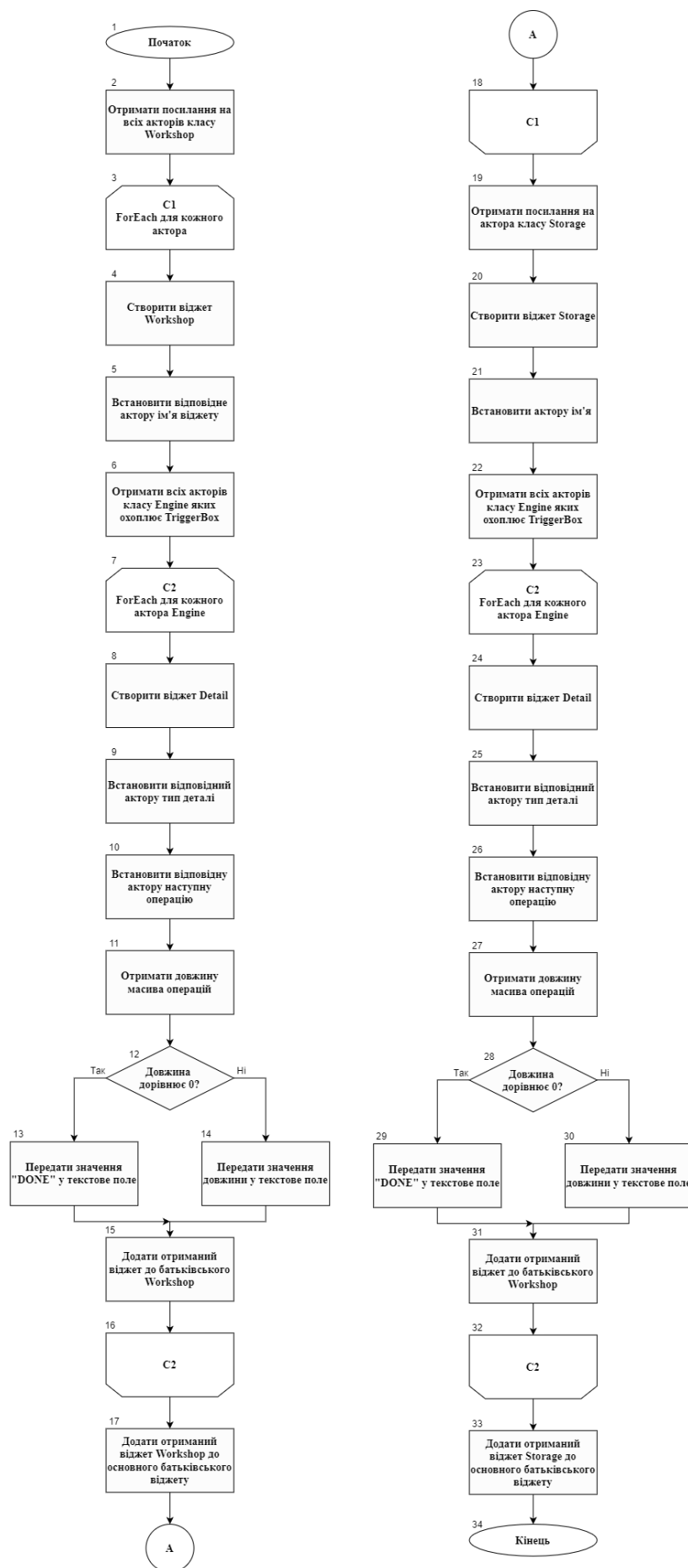


Рисунок 3.12 – Блок-схема алгоритму відображення статусу деталей

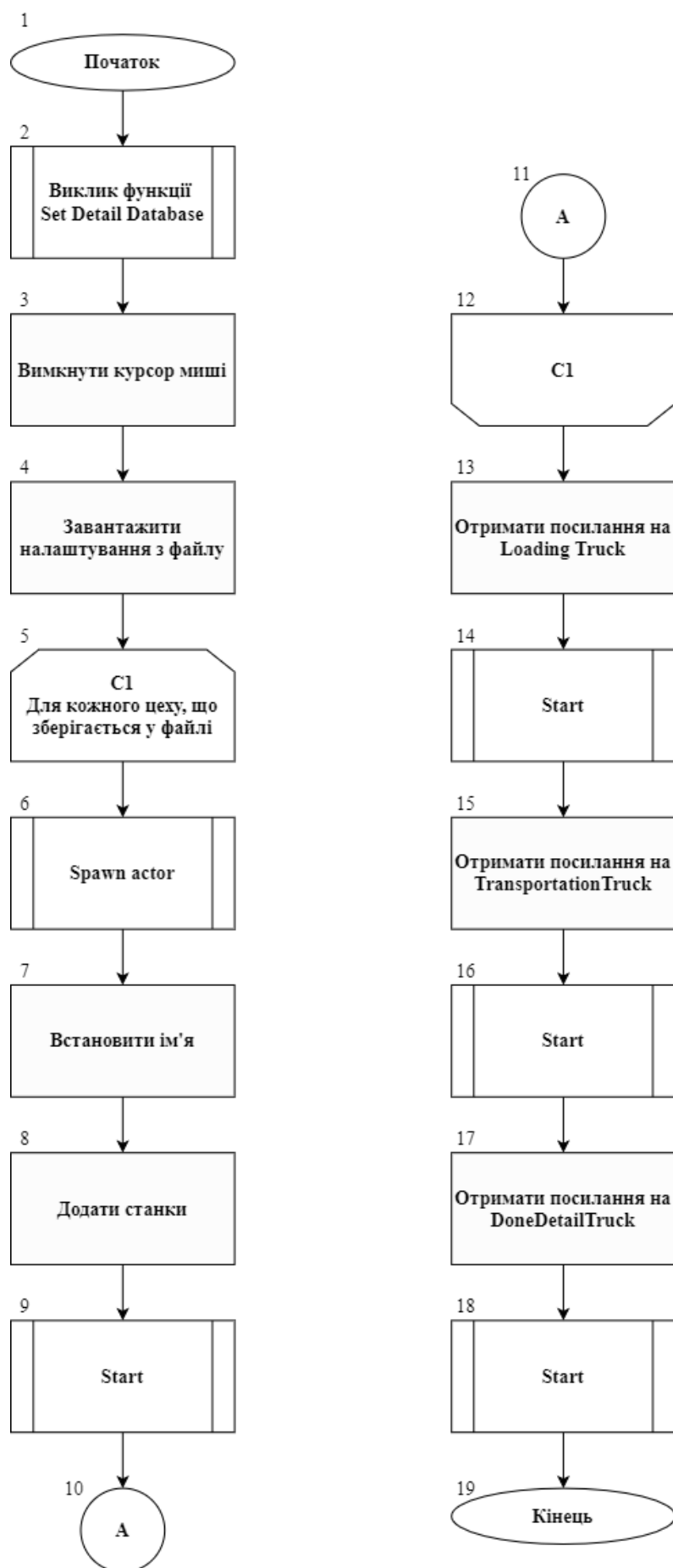


Рисунок 3.13 – Блок-схема роботи алгоритму запуску симуляції



### Функціональність кнопки «Change settings»

Система встановлення налаштувань для подальшої симуляції складається з 4-ох стадій:

1. Введення назв всіх операцій для деталей;
2. Встановлення типів деталей, набору введених операцій для кожної з них, загальної кількості для обробки кожного типу;
3. Встановлення кількості цехів, кількості станків всередині кожного цеха, операцій, що повинен виконувати кожен з станків для деталей та часу, що займає кожна операція для кожного станка;
4. Система генерації та позиціювання кожного цеху у відносних координатах світу на площині.

Після натискання клавіші «Change settings» спочатку завантажується основний віджет – віджет керування стадіями встановлення налаштувань, що є незмінним на протязі перших 3-х стадій встановлення налаштувань (рис. 3.14). Пізніше до його верхньої частини додаються віджети встановлення налаштувань перших 3-х стадій.

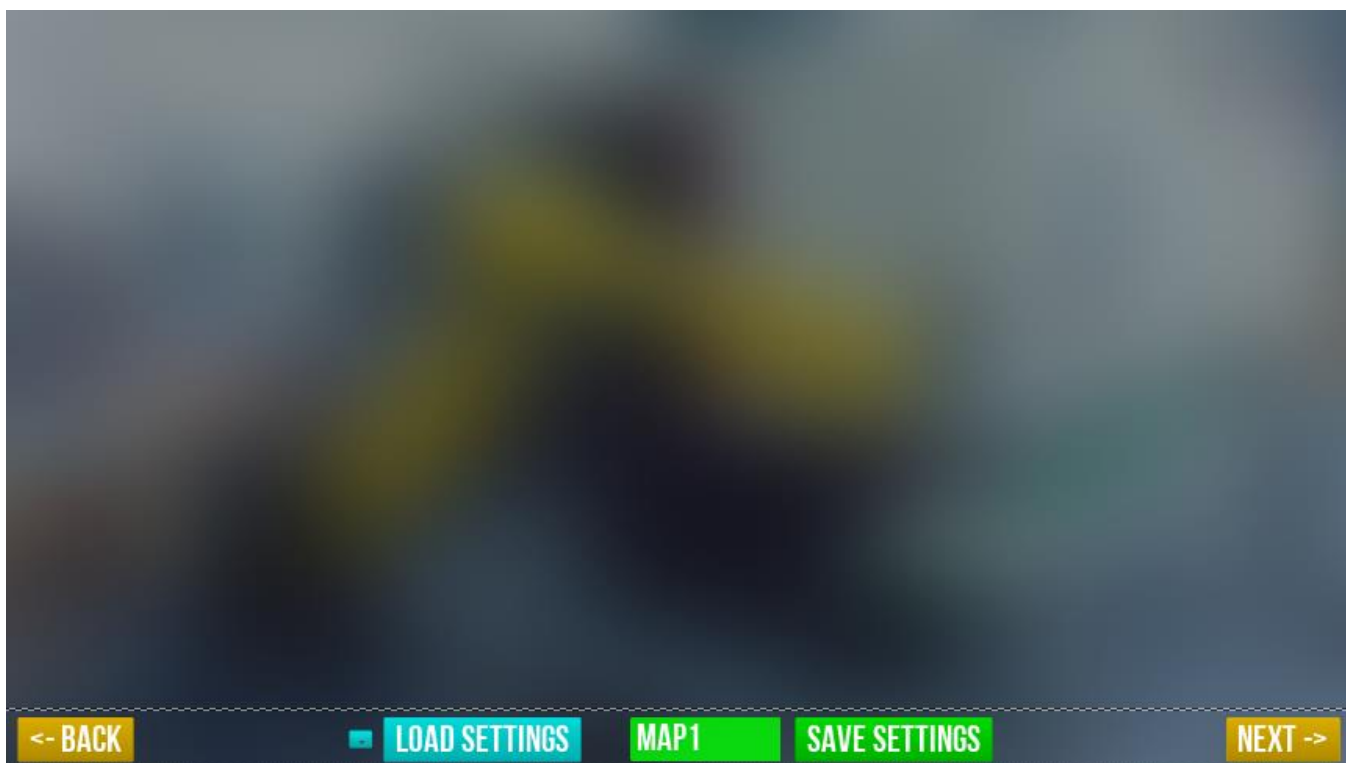


Рисунок 3.14 – Віджет керування стадіями встановлення налаштувань

Логіка віджету керування стадіями налаштувань є окремою від основних віджетів стадій.



Рисунок 3.15 – Блок-схема алгоритму завантаження віджету та блок-схема алгоритму керування введенням текстом в поле для збереження файлів

Одразу після завантаження віджет керування стадіями додає віджет встановлення налаштувань 1-ої стадії до батьківського віджету. Також завантажуються імена існуючих файлів, в яких раніше було збережено налаштування. За допомогою цикла `ForEach` імена додається до опцій `ComboBox`.

Якщо була натиснена кнопка «Back» одразу перевіряється, на якій стадії зараз знаходиться користувач. Якщо це 1-ша стадія, то віджет, що в даний момент знаходиться на екрані приховується, а віджет головного меню додається на екран. Якщо це 2-га стадія, то віджет 2-ої стадії видаляється від батьківського віджета, а замість нього до батьківського віджета додається віджет 1-ої стадії, та у змінну встановлюється статус «1-а стадія». Якщо це 3-тя стадія, то віджет 3-ої стадії видаляється від батьківського віджета, а замість нього до батьківського віджета додається віджет 2-ої стадії, та у змінну встановлюється статус «2-а стадія».

Якщо була натиснута кнопка «Next», спочатку перевіряється, на якій стадії встановлення налаштувань знаходиться користувач:

1. Якщо це 1-ша стадія, то алгоритм перевіряє, чи більше 2-ох операцій додав користувач. Якщо операцій менше 2-ох, то до екрану додається віджет «Error», який повідомляє, що перед переходом до наступної стадії користувач повинен додати в список мінімум 2 операції. Якщо операцій більше 1, то віджет стадії 1 видаляється від батьківського віджета. Після цього перевіряється, чи вже існує віджет стадії 2. Якщо віджет стадії 2 ще ніколи не був відображений, то алгоритм створює віджет стадії 2, встановлює посилання на нього та додає до батьківського віджету. Якщо віджет вже існує, то за допомогою посилання він просто додається до батьківського віджету. Потім встановлюється статус «стадія 2».
2. Якщо це 2-а стадія, перевіряється, чи існує посилання на віджет 3-ої стадії. Якщо посилання існує, то віджет 2-ої стадії видаляється від батьківського віджета, а віджет 3-ої стадії через посилання додається до батьківського віджета. Якщо віджет раніше не був створений, то віджет 2-ої стадії видаляється від батьківського віджета, створюється віджет 3-ої стадії, створюється

посилання на нього та додається до батьківського віджета. Встановлюється статус «стадія 3».

3. Якщо це 3-я стадія, то за допомогою цикла `ForEach` до масиву записуються всі унікальні обрані операції для деталей на стадії 2, до іншого масива за допомогою цикла `ForEach` записуються всі унікальні обрані операції для станків на стадії 3. Після цього, за допомогою цикла `ForEach` по унікальним операціям для деталей перевіряється, чи всі обрані на 2-ій стадії деталі зможуть успішно пройти цикл обробки через додані на 3-ій стадії станки. Якщо всі деталі не зможуть успішно пройти цикл обробки, на екран додається віджет «Error», який повідомляє користувача, що неможливо перейти до останньої стадії, оскільки станки повинні містити всі операції, які потрібні деталям. Віджет зникає після натискання користувачем кнопки «OK». Якщо після перевірки для всіх деталей існує хоч один станок який виконує потрібну операцію, то до екрану додається віджет «Question», який запитує у користувача, чи впевнений він що хоче перейти до наступних стадій з обраними налаштуваннями. Після стверджувальної відповіді від користувача, алгоритм додає до екрану віджет «Name slot», який запитує користувача, яке ім'я обрати для файла, в якому буде збережено налаштування. Після обрання користувачем коректного ім'я файла, запускається функція «Save», що зберігає у заданий файл налаштування з попередніх 3-х стадій. Після цього віджет 3-ої стадії видаляється від батьківського віджета, також видаляється батьківський віджет керування стадіями встановлення налаштувань та завантажується рівень «CustomizationMap».

Кнопка «Save» відповідає за збереження вже встановлених налаштувань в файл, ім'я якого вказує користувач у текстовому полі ліворуч від самої кнопки «Save». Якщо по введеному імені буде знайдено вже існуючий файл, програми запитає користувача, чи бажає він перезаписати вказаний файл. При позитивній відповіді старий файл буде видалено, а замість нього додано новий з новими налаштуваннями і таким самим ім'ям. Одразу після натискання кнопки «Save» та визначення існування такого файлу, запускається функція «Save».

Функція «Save» створює об'єкт класу Save Object, а далі починає зберігати в нього поступово кожен етап встановлення налаштувань. Запускається цикл ForEach для кожного посилання на глобальні операції, що були створені на 1-й стадії налаштувань. За допомогою посилання з операції зчитується ім'я та записується у масив «Op Name Set» у Save Object.

Далі перевіряється, чи існує посилання на віджет 2-ої стадії налаштувань. Якщо посилання існує, то запускається цикл ForEach для кожного посилання на тип деталі, що був створений на 2-й стадії налаштувань. Для кожної з деталей запускається цикл ForEach за посиланням на ComboBox кожної доданої операції до цієї деталі. З кожного ComboBox зчитується обрана опція та записується до масиву формату String «Selected Options». Коли всі імена доданих операцій записані, програма створює структуру даних, в яку записуються: ім'я (тип) деталі, масив «Selected Options», кількість (популяція) деталей та посилання на деталь. Після цього, створена структура додається до масиву структур даних деталей та звільняється масив «Selected Options». Після проходження двох циклів ForEach, масив структур даних деталей записується в об'єкт Save Object та звільняється від даних.

Далі перевіряється, чи існує посилання на віджет 3-ої стадії налаштувань. Якщо посилання існує, то запускається цикл ForEach для кожного посилання на цех, що був створений на 3-й стадії налаштувань. Перед цим звільняється масив структур даних станків. Для кожного цеху запускається цикл ForEach за посиланням на віджет кожного станка. З кожного віджета зчитується та записується у структуру даних: ім'я станка, обрана опція ComboBox (операція яку він виконує), час що займає виконання операції. Отримана структура записується до масиву структур даних «Machine Set Struc». Коли посилання на всі станки оброблені, програма створює структуру даних, в яку записуються: ім'я цеху, масив «Machine Set Struc» та посилання на цех. Після цього, створена структура додається до масиву структур даних цехів що має назву «Workshop Set Struc». Після проходження двох циклів ForEach, масив структур даних цехів «Workshop Set Struc» записується в об'єкт Save Object, після чого «Machine Set Struc» і «Workshop Set Struc» звільняються від даних.

Після проходження вищеописаних алгоритмів, створений об'єкт Save Object зберігається у файл. З іншого файлу завантажується масив імен всіх файлів збережених налаштувань, до нього додається новий, щойно створений файл. Якщо ім'я файлу було унікальне, до ComboBox віджету керування стадіями встановлення налаштувань додається нова опція та обирається ім'я новоствореного файлу. Блок-схема вищеописаної функції представлена у додатку.

Якщо натиснута кнопка «Load», програма спочатку перевіряє, чи пустий ComboBox був перед її натисканням. Якщо так, то користувачу виводиться повідомлення за допомогою віджета «Error». Якщо ні, то до дії приступає алгоритм завантаження налаштувань. Завантажуються налаштування з обраного у ComboBox файла. Перед завантаженням налаштувань всіх стадій, алгоритм звільняє всі ComboBox на 2-ій та 3-ій стадіях налаштувань від обраних функцій.

1. Після цього, на 1-ій стадії налаштувань очищаються усі раніше додані віджети операцій, та звільняється масив посилань на них. Запускається цикл ForEach для кожного імені операції з завантаженого файла. В процесі циклу, функцією «Add Operation» (рис. 3.14) 1-ої стадії налаштувань поступово додаються всі збережені операції. Далі перевіряється статус стадії налаштувань, на якій зараз знаходиться користувач.
2. Якщо це 2-а стадія налаштувань, то алгоритм перевіряю, чи існують посилання на віджет 2-ої стадії налаштувань. Якщо ні, то він негайно створюється та записується у посилання, якщо так, то запускається цикл ForEach по кожному посиланню на віджет операції у деталях, та очищаються усі опції у ComboBox. Потім у самому віджеті звільняється від дитячих елементів поле, що заповнюється віджетами деталей. З об'єкта Save Object зчитується масив структур «Detail Set» та запускається цикл ForEach по його елементам. В процесі циклу, функцією «Add Detail» 2-ої стадії налаштувань, додається деталь, та за допомогою наступного циклу ForEach та зчитаного масиву «Selected operations» до кожної деталі, за допомогою функції «Add Operation to Detail» (рис. 3.16), додається збережена кількість операцій та у кожному ComboBox обираються опції, що також були збережені. Після завершення

циклу по операціях, у текстове поле заповнюється інформація про популяцію цього типу деталі.

Для останньої, 4-ої стадії встановлення налаштувань існують інші алгоритми, що будуть описані далі.



Рисунок 3.16 – Блок-схема роботи функції «Add Operation to Detail»

### Перша стадія встановлення налаштувань – введення операцій

На першій стадії користувач повинен ввести кодові назви для всіх операцій, що надалі будуть використовуватися для позначення операцій, які виконують станки та проходять деталі. Таких операцій має бути не менше двох. Користувачу доступне поле для введення кодових назв операцій, зелена кнопка зчитування текстового поля та додавання операції та червона кнопка видалення для кожної з введених операцій (рис 3.12).

### Друга стадія встановлення налаштувань – встановлення типів деталей

На другій стадії налаштувань користувач може створювати різні типи деталей з різним набором операцій для кожної (рис. 3.20). В кожен ComboBox деталі додаються операції, введені на першій стадії налаштувань. В кінці кожного рядку операцій міститься поле «Population», в яке можна ввести бажану кількість деталей що будет згенеровано та оброблено. Коли користувач натискає клавішу «+» знизу, додається новий тип деталі за допомогою функції «Add Detail», її алгоритм описано на блок-схемі (рис. 3.21).

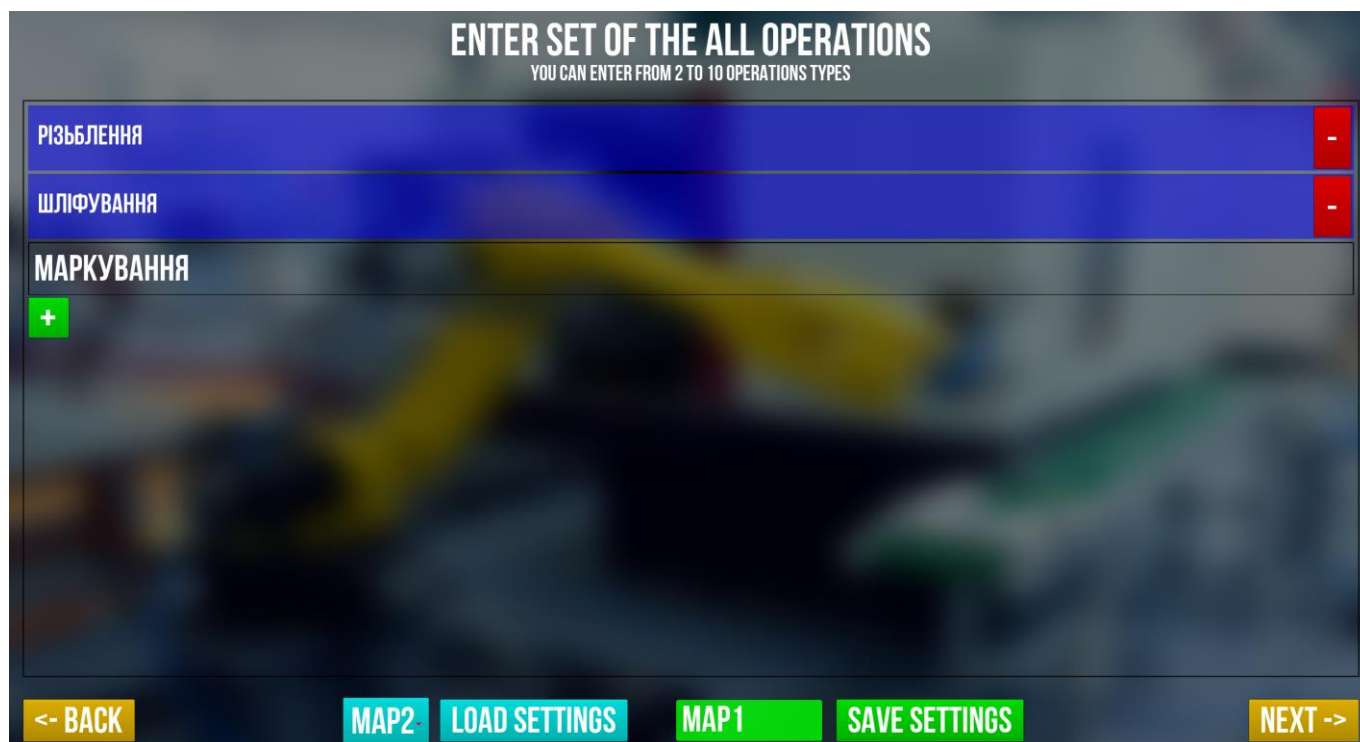


Рисунок 3.17 – Вікно першої стадії - введення операцій



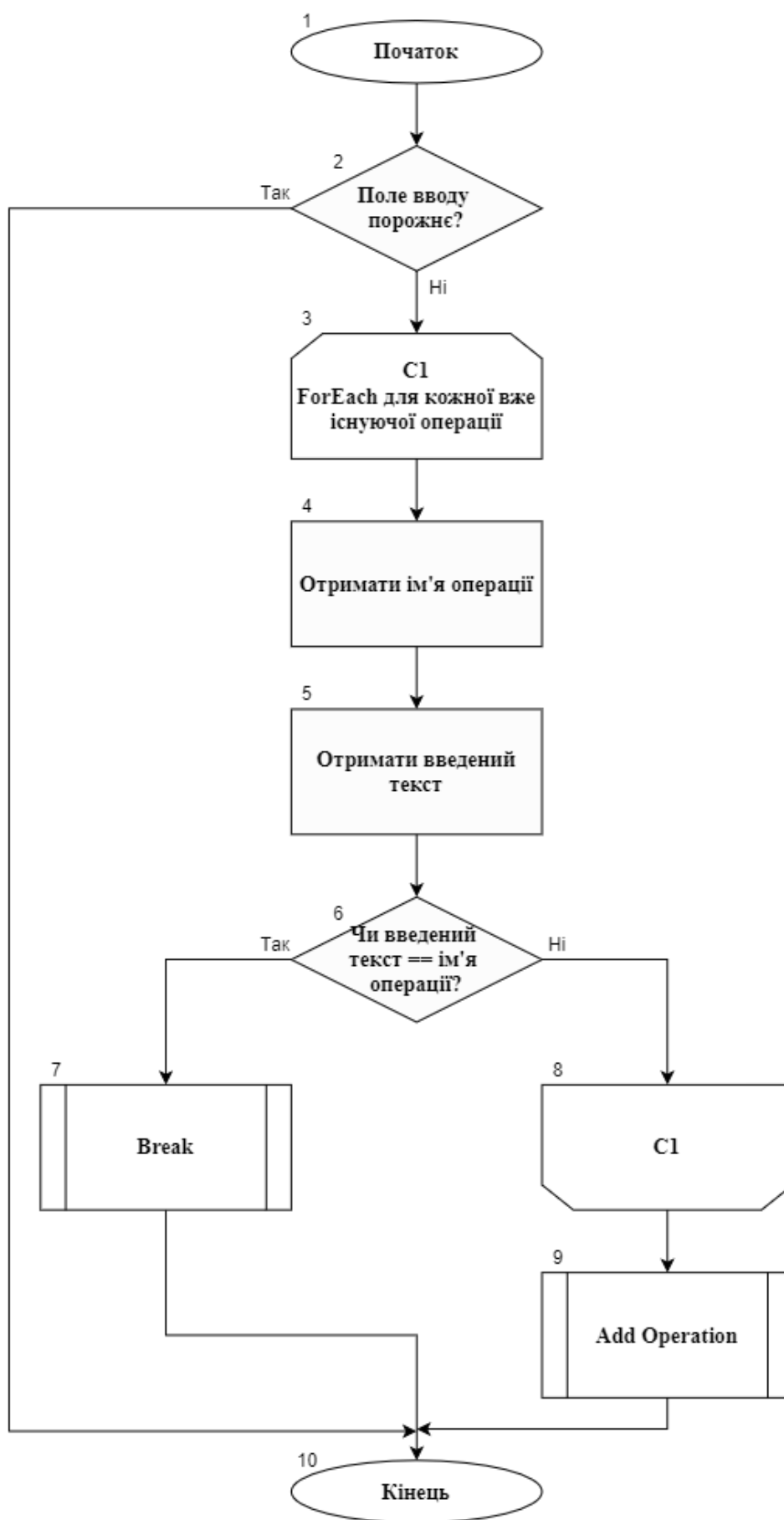


Рисунок 3.18 – Блок-схема роботи алгоритму введення операції (реакція програми на натискання кнопки «Плюс»)

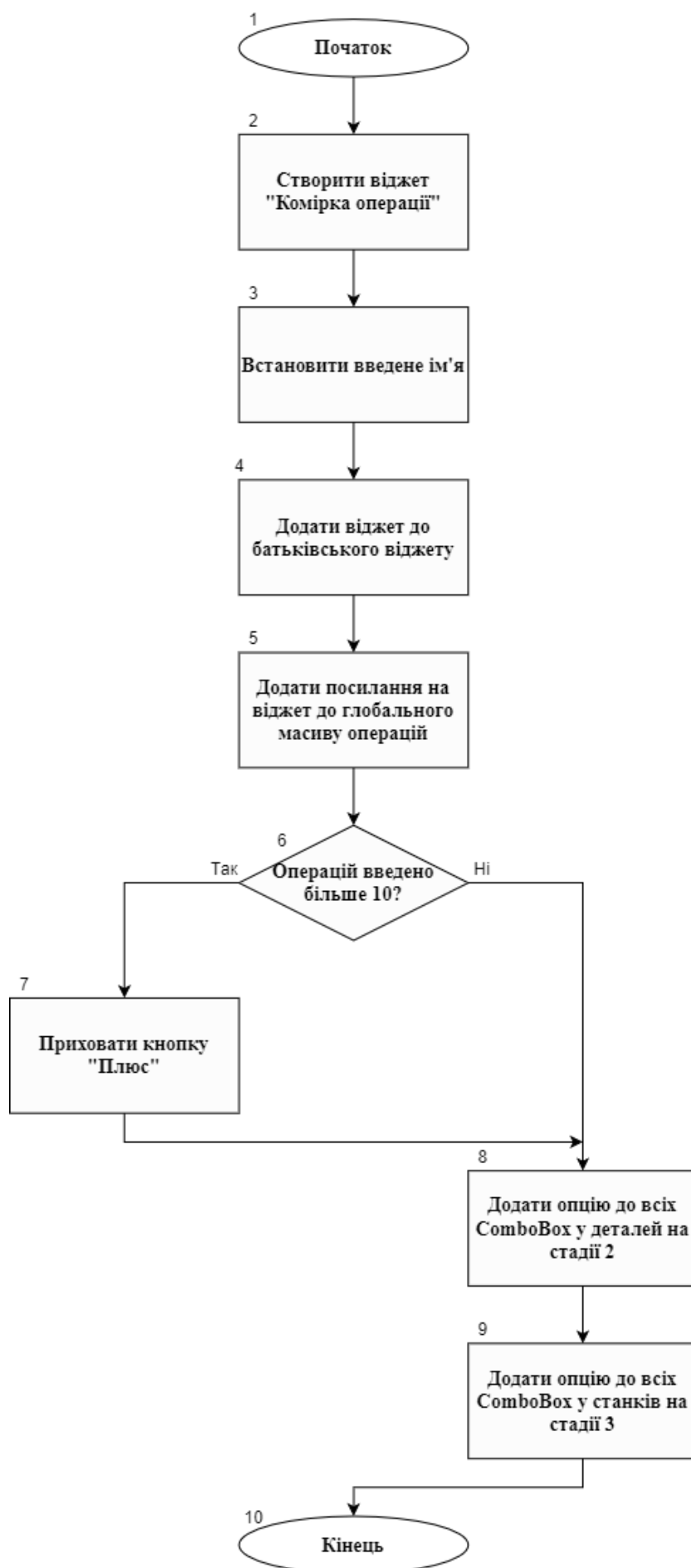


Рисунок 3.19 – Блок-схема алгоритму функції «Add Operation»

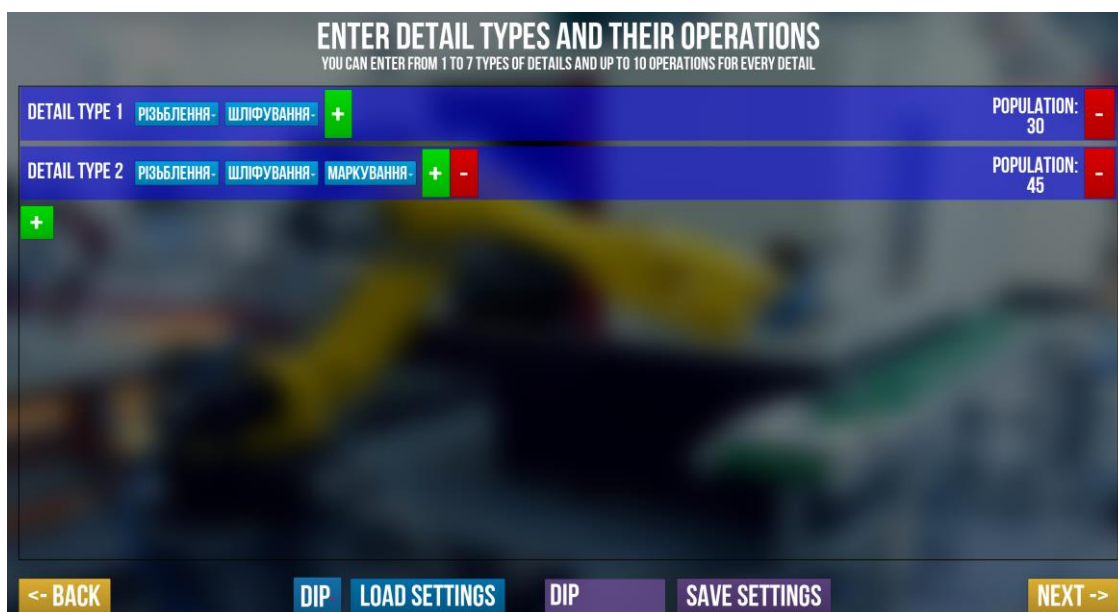


Рисунок 3.20 – Вікно другої стадії встановлення налаштувань

### Третя стадія встановлення налаштувань – встановлення цехів

На третій стадії встановлення налаштувань користувач повинен створити від одного до семи цехів, в кожному з яких повинно бути по 2 станки мінімум. В кожен ComboBox станків додаються операції, введені на першій стадії налаштувань. Також для кожного з станків користувач вказує час, потрібний для обробки деталі цією операцією. Час рахується у секундах, та може бути від 1 секунди до 180 (рис. 3.22). Цехи додаються по натисканню на клавішу «+» за допомогою функції «Add Workshop» (рис 3.23).

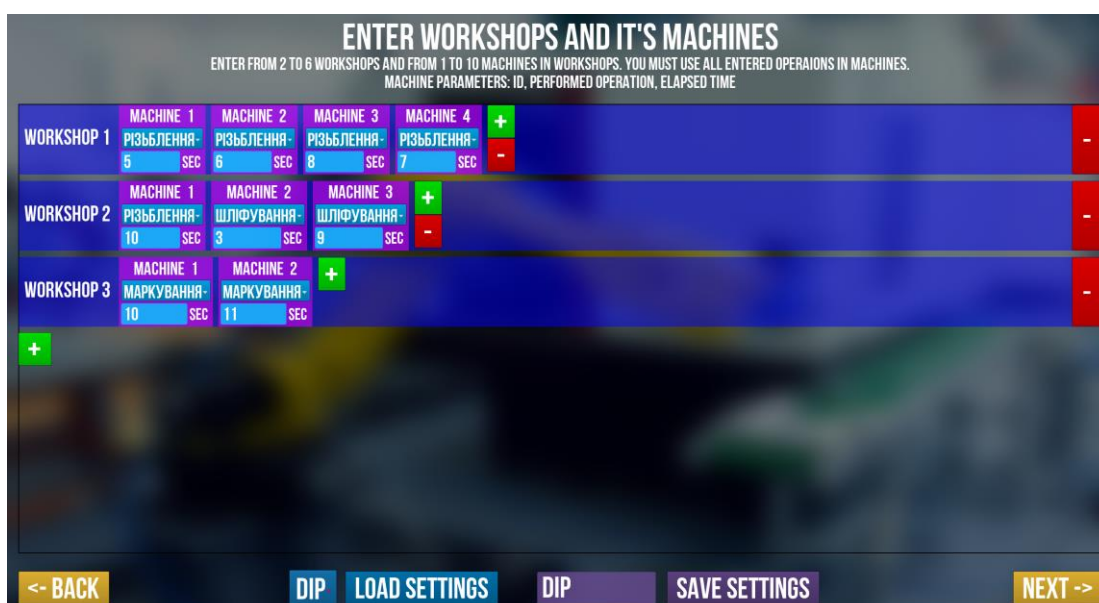


Рисунок 3.22 – Вікно третьої стадії встановлення налаштувань

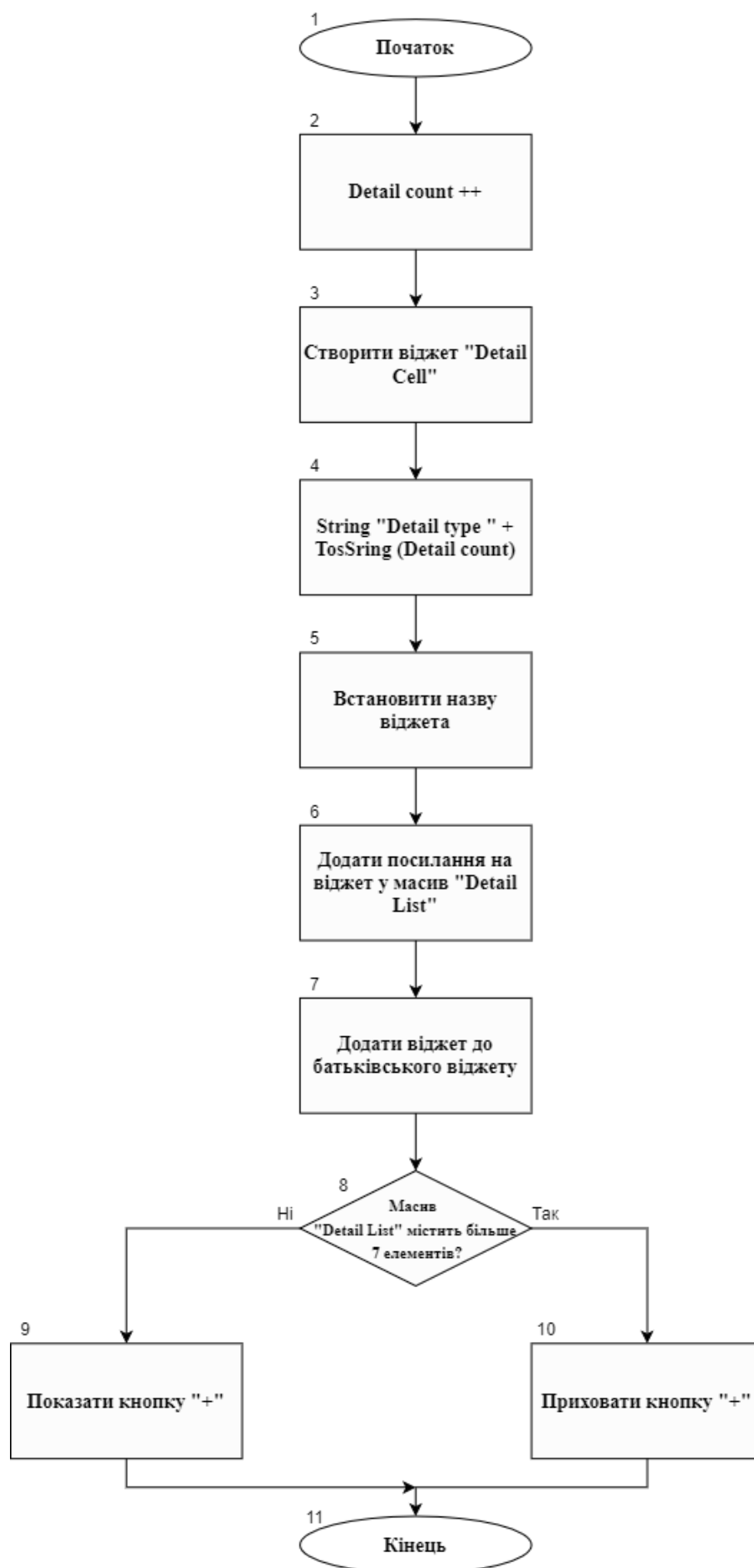


Рисунок 3.21 – Блок-схема алгоритму функції «Add Detail»

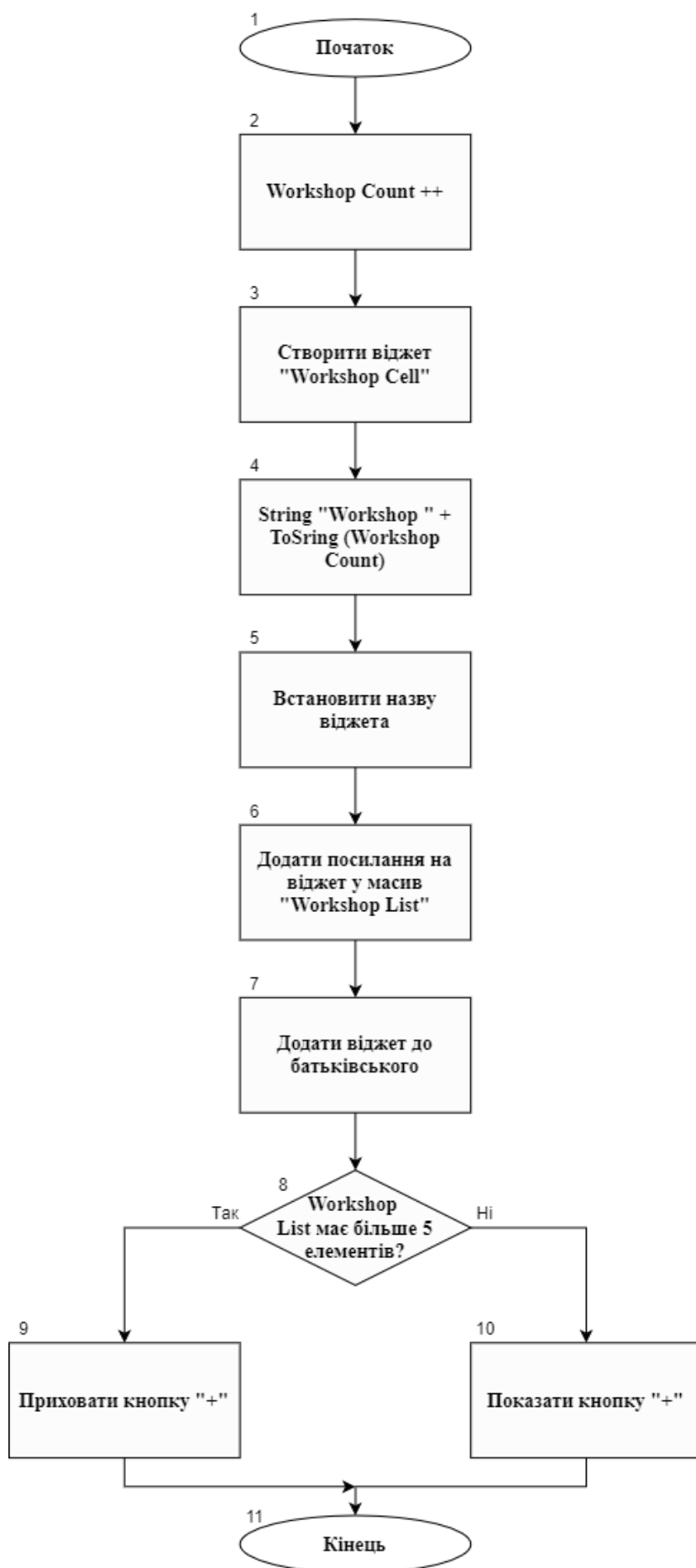
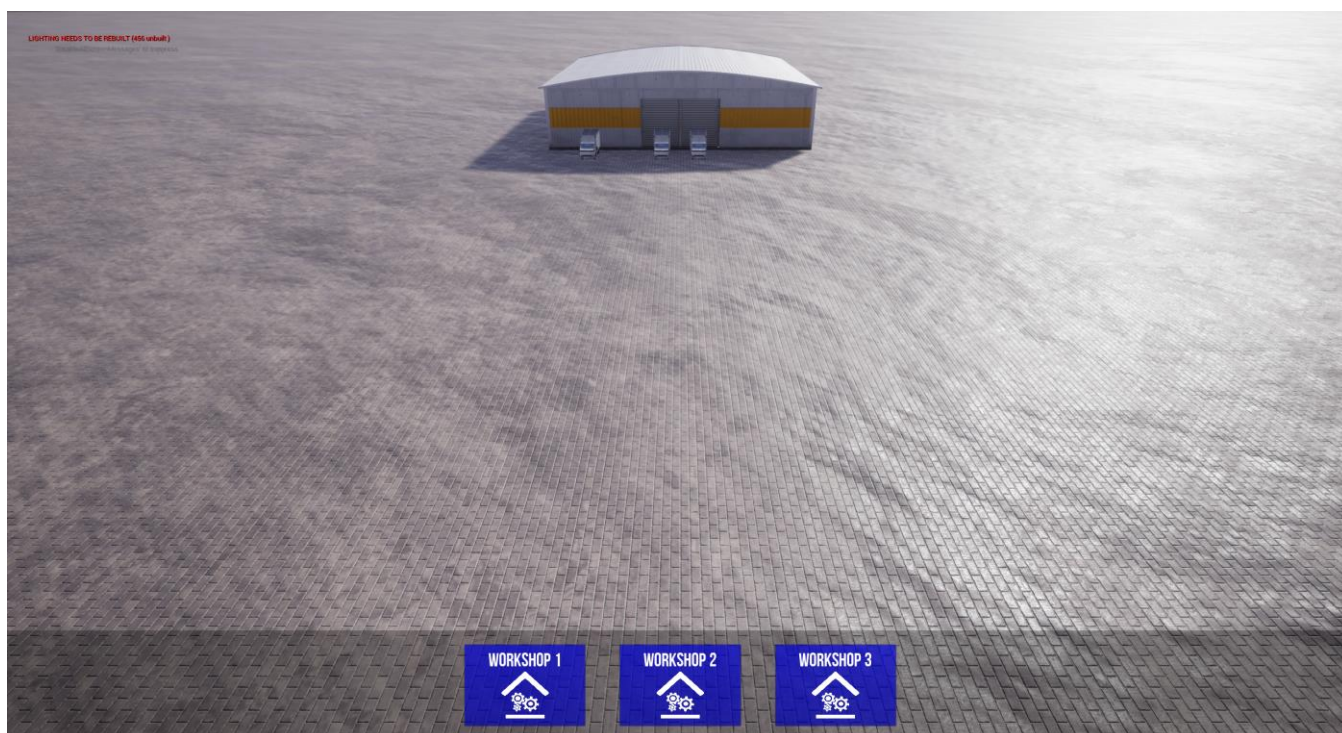


Рисунок 3.23 – Блок-схема алгоритму функції «Add Workshop»

### Четверта стадія встановлення налаштувань – розташування цехів

В четвертій і останній стадії встановлення налаштувань користувач має встановити всі створені на 3-ій стадії цехи в ті місця на площині, де він бажає їх бачити. Перше що побачить користувач після переходу до цієї стадії – це центральний склад та список створених ним цехів знизу (рис. 3.24).



*Рисунок 3.24 – Початкове вікно четвертої стадії встановлення налаштувань*

Щоб обрати цех та почати керувати його встановленням, користувач повинен натиснути на будь-яку кнопку знизу, на яких намальовані умовні позначення цехів та їх відповідна назва. Після натиснення на одну з кнопок, камера користувача плавно переміститься у початкову точку чітко позаду цеха, та керування цехом стане доступне. Клавiші «W», «A», «S», «D» або «Up», «Down», «Left», «Right» відповідають за переміщення цеху вздовж умовних осей X та Y на площині. Клавiші «V» та «B» відповідають за зміну швидкості переміщення. Індикатор швидкості переміщення виводиться у лівому верхньому кутку екрана, одразу під індикаторами умовних координат та кута повороту (рис. 3.25). Підказка зі всіма ключовими клавiшами керування виведена справа в верхній частині екрана. Клавiші «Z» та «X» відповідають за обертання цеху навколо осі Z. Натиснення клавiші «CTRL» показує або приховує курсор миші, який потрібний для переключення керування між цехами.

За допомогою миші користувач може обертати камеру навколо цеху, щоб мати зручний огляд місцевості, а за допомогою колеса миші може редагувати відстань (радіус обльоту) камери від цеху, що знаходиться під керуванням.



*Рисунок 3.24 – Процес встановлення цеху під час четвертої стадії*

Як тільки користувач встановить останній цех (для цього потрібно хоча б раз натиснути на кожну кнопку що прив'язані до цехів) у правій нижній частині екрану з'явиться клавіша «Start simulation», натиснувши яку користувач зможе запустити симуляцію на основі зроблених налаштувань (рис. 3.25).

Якщо клавіша бує натиснена, автоматично виконається функція збереження всіх налаштувань «Save», блок-схема алгоритму якої є в додатку. Далі, на основі збережених налаштувань, буде запущено симуляцію – у всіх акторів сцени будуть запущені алгоритми «Start», що відповідають за початок аналізу отриманих даних та початок роботи всіх об'єктів. Камера користувача автоматично переміститься у початкове положення та отримає статус «камери спостерігача», що дозволить користувачу вільно переміщувати камеру та слідкувати за процесом симуляції (рис. 3.26).



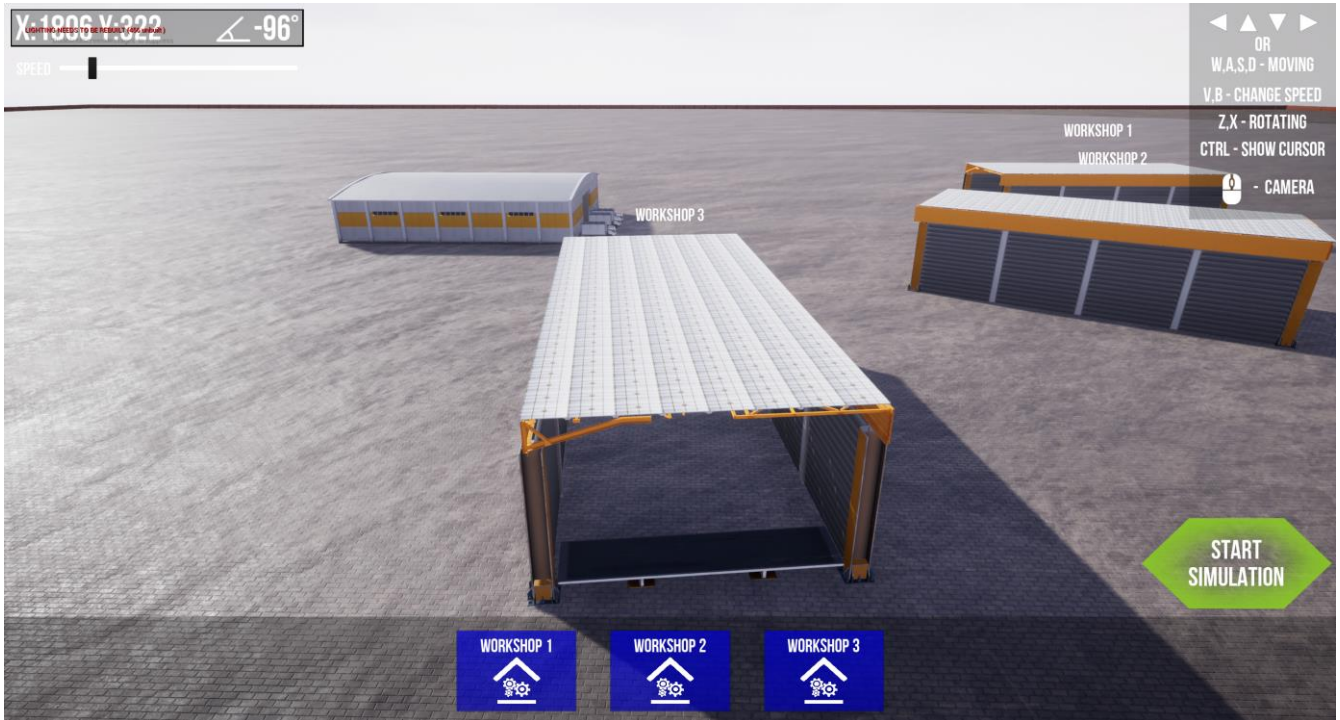


Рисунок 3.25 – Закінчення процесу встановлення цехів та відображена клавіша «Start simulation»

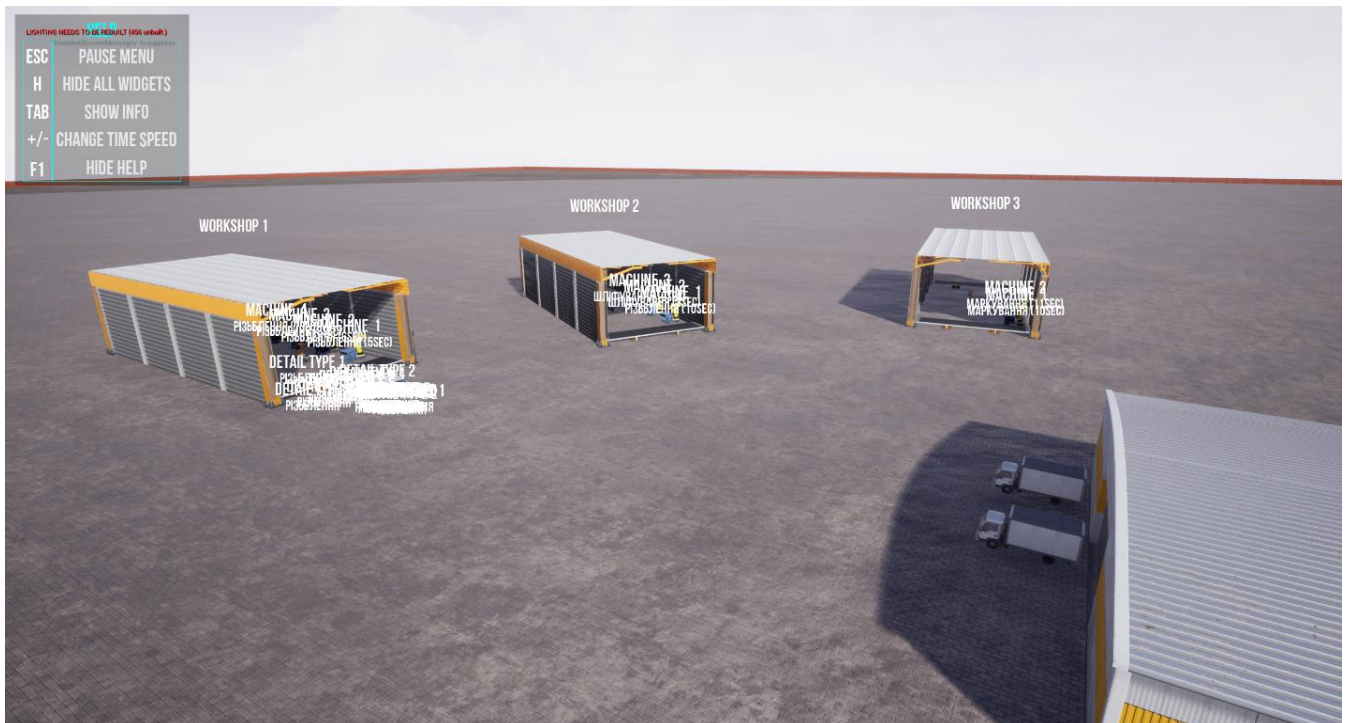


Рисунок 3.26 – Результат натиснення клавіші «Start simulation» після встановлення налаштувань



### Функціональність кнопки «Save file manage»

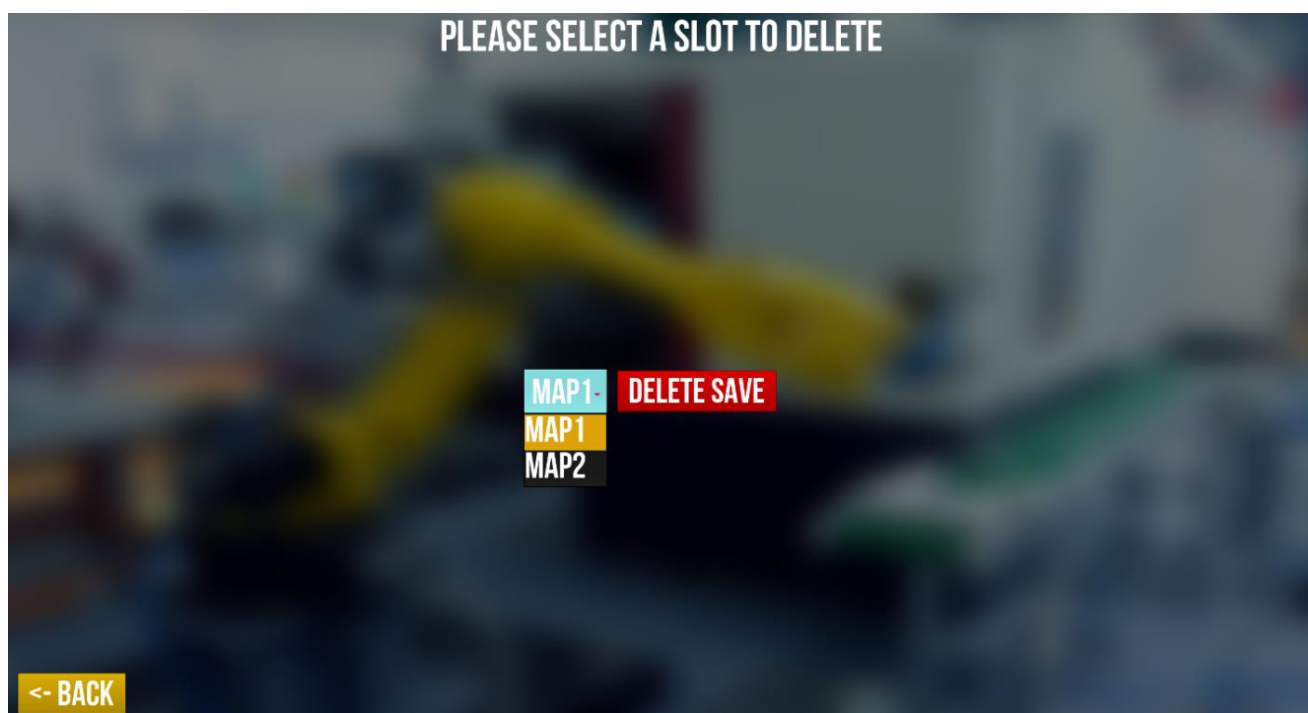


Рисунок 3.27 – Вікно *Save file manage*

Після натиснення клавіші «Save file manage» користувачу відкривається окреме вікно (рис. 3.27) де, можна обрати один з раніше збережених файлів налаштувань та видалити його. Після натиснення клавіші «Delete», програма запитає, чи впевнений користувач, що хоче видалити обраний слот (рис 3.28).

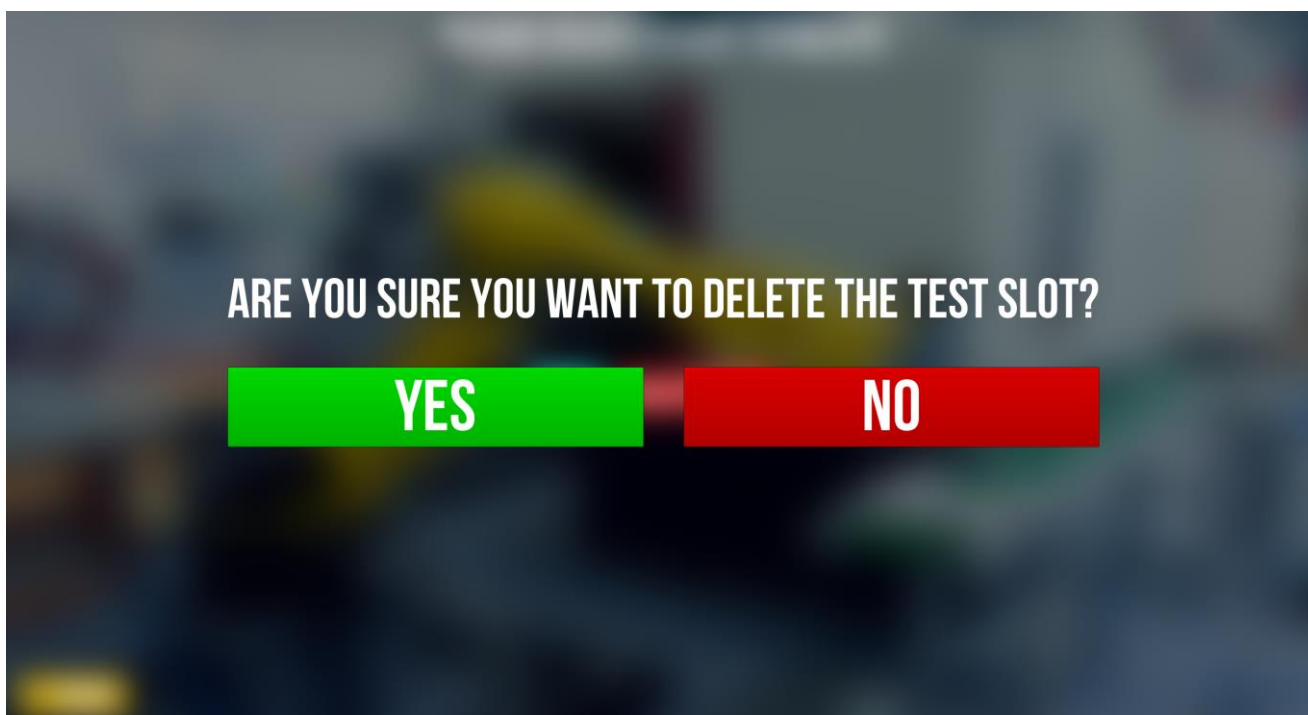


Рисунок 3.28 – Вікно з питанням користувачу

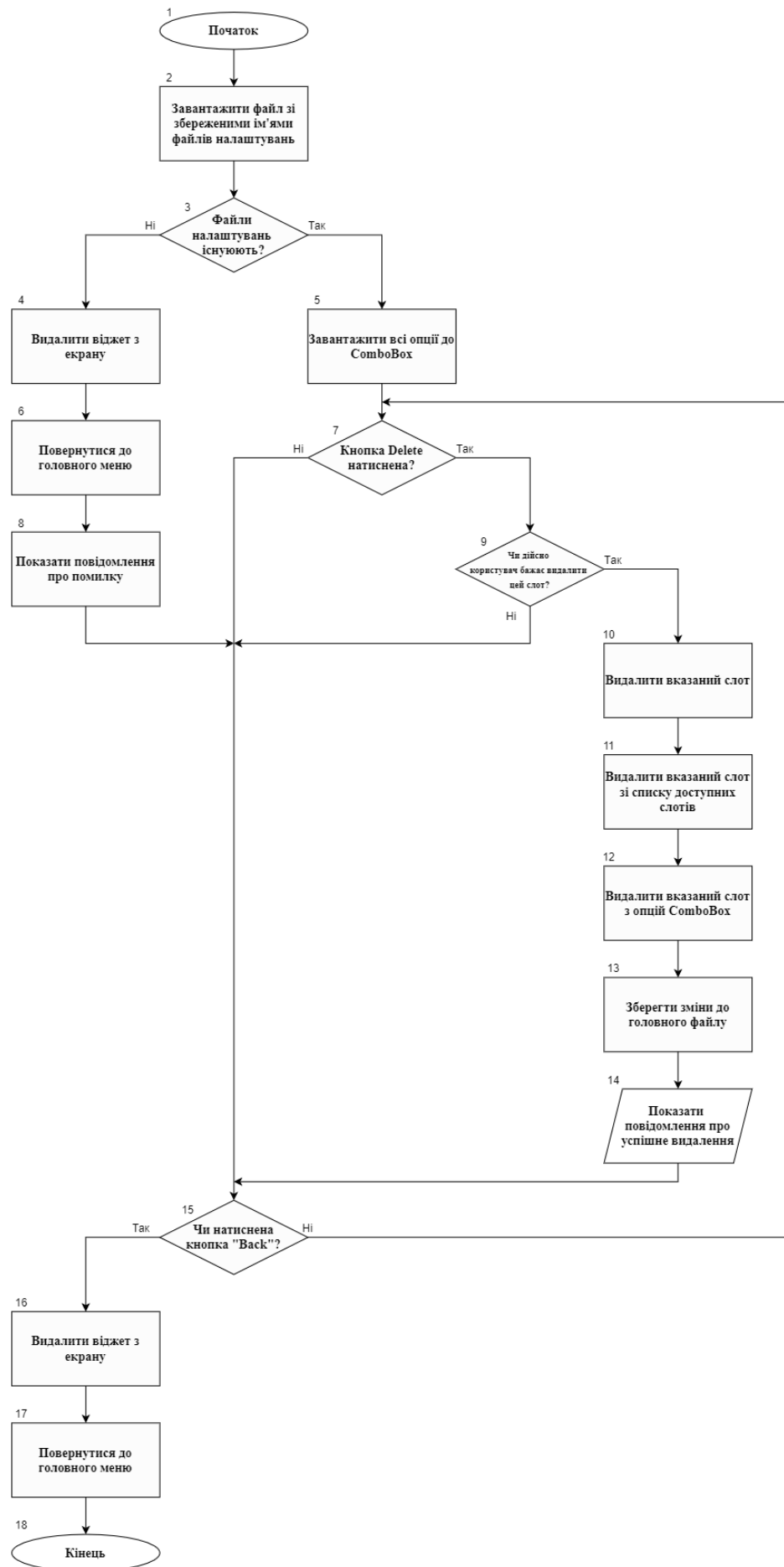
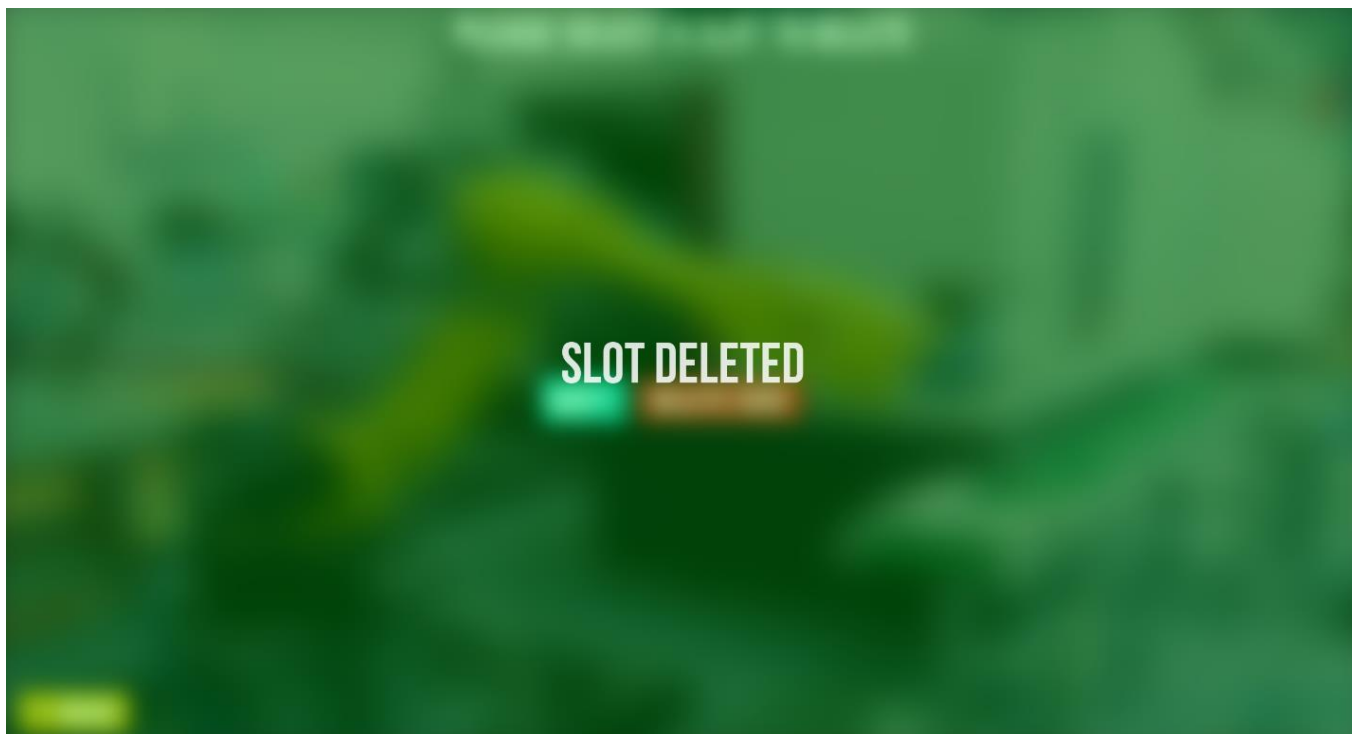


Рисунок 3.30 – Блок-схема логіки роботи вікна «Save file manage»

Після підтвердження видалення користувачем, посеред екрану з'являється напис «Slot Deleted» – «Слот видалено». Він поступово автоматично зникне (рис. 3.29).



*Рисунок 3.29 – Вікно, що повідомляє про успішне видалення слота зі збереженими налаштуваннями*

### **Деталь (Detail)**

Деталь – це об’єкт у середовищі, який зберігає набір операцій що треба пройти та назву, яка відображається у віджеті над деталю. Після того як буде розпочато симуляцію, з введених користувачем та збережених даних запитом отримуються типи деталей та кількість кожного типу. Потім, за допомогою циклу ForEach, що запускається по загальній кількості всіх деталей, створюється база даних, в яку додається чітка кількість кожного типу деталей. Коли вантажівка генерує деталі, вона звертається до цієї бази даних та випадково обирає з неї потрібну для завантаження кількість деталей. Кожному типу деталі відповідає певний набір операцій, що ввів користувач. Над кожною деталлю відображається віджет, що містить коротку інформацію про деталь, а саме: «Detail type X» – де X – це ID типу деталі та одразу нижче назву операції, яку деталі треба пройти наступною.

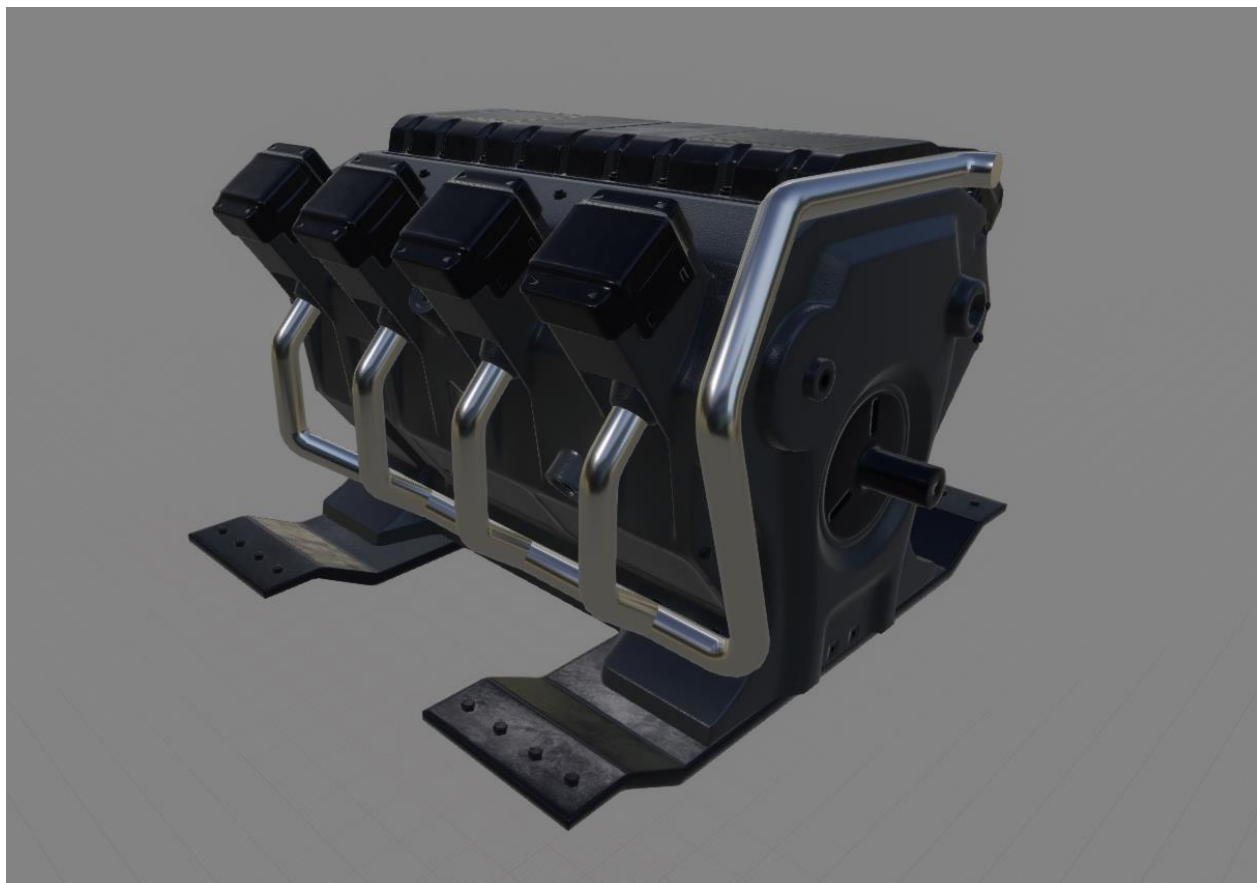


Рисунок 3.31 – 3D-модель деталі

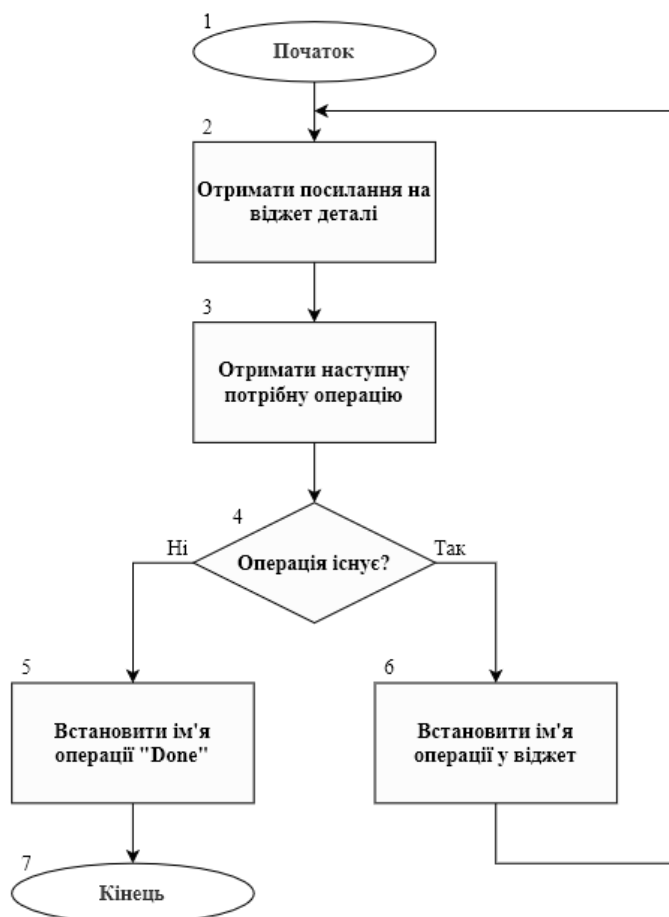


Рисунок 3.32 – Блок-схема алгоритму Detail

### Автономний транспортний модуль (АТМ)

АТМ – невід'ємна частина кожного цеху. АТМ відповідає за транспортування деталей, що підлягають обробці до станків, за транспортування деталей між станками та за транспортування та складання деталей що вже пройшли обробку (рис. 3.33).



*Рисунок 3.33 – 3D-модель АТМ що використовується в середовищі*

Після старту симуляції, алгоритм АТМ звертається до цеху в якому знаходиться, та щосекунди перевіряє, чи є у черзі деталі що підлягають обробці. Якщо черга має посилення на деталі, алгоритм обирає встановлений індекс з масиву та перевіряє його валідність. Якщо індекс невалідний, встановлюється значення 0 та очікується 1 секунду. Після чого алгоритм знову повертається до черги з деталей. Якщо індекс валідний, алгоритм очищає масив потенційно відповідних станків, встановлює посилення на першу деталь з черги у змінну «Target Detail» та запускає цикл ForEach для кожного станка у цеху. У циклі перша потрібна деталі операція порівнюється з операцією що виконує станок. Якщо станок може обробити деталь, посилення на нього додається до масиву потенційно відповідних станків. Після завершення циклу, алгоритм перевіряє, чи пустий масив потенційно відповідних станків, та діє відносно отриманого результату:

1. Масив має елементи. Запускається цикл ForEach для кожного елементу масива потенційно відповідних станків. В циклі перевіряється, чи вільний в даний

момент станок. Якщо ні – він видаляється з масиву. Якщо після циклу масив потенційно відповідних станків виявляється пустим, то інкрементується індекс обрання з черги та алгоритм повертається на момент обрання деталі з черги цеху. Якщо масив не пустий, то запускається новий цикл ForEach для кожного елемента масиву потенційно відповідних станків. У тілі циклу порівнюється час виконання операцій у кожного станка та видаляються станки у яких час більший. Після завершення циклу, запускається новий цикл ForEach для кожного елемента масиву потенційно відповідних станків. У тілі циклу порівнюються відстані від деталі до станків та обирається станок з найменшою відстанню. Після завершення циклу, звільняється від елементів масив потенційно відповідних станків, та АТМ починає рух до деталі, посилання на яку знаходиться у змінній «Target Detail». Після досягнення цілі, відбувається затримка 1 секунду, деталь переміщується на платформу АТМ, встановлюється відповідний кут обороту та модель деталі прив'язується до АТМ. Алгоритм перевіряє, чи деталь після обробки на станку, якщо ні, то позначає місце де була деталь вільним. Деталь видаляється з черги на обробку у цеху та АТМ починає рухатись до цілі, посилання на яку знаходиться у «Tager Machine». Після досягнення цілі відбувається затримка 1 секунду, деталь відв'язується від АТМ, переміщується на платформу станка та встановлюється кут повороту як у станка. Після цього алгоритм повертається у початок, на момент перевірки черги деталей на обробку.

2. Масив пустий. АТМ починає рухатися до цілі «Target Detail», після її досягнення відбувається затримка довжиною в 1 секунду. Потім АТМ переміщує деталь на свою платформу, встановлює відповідний собі кут обороту та прив'язує модель деталі до себе. Деталь видаляється з черги на обробку та запускається цикл ForEach для кожного місця що позначене як місце для деталей що готові до транспортування. Якщо всі місця зайняті, АТМ очікує 1 секунду та знову виконує цикл. Якщо ж вільне місце знаходиться, цикл переривається за допомогою програми Break, виявлене місце позначається як наступна ціль та зміню свій статус на «Зайняте». АТМ починає рухатися до цілі,

одразу після її досягнення відбувається затримка на 1 секунду, потім деталь відв'язується від АТМ, переміщується на відповідне місце, встановлює кут оберту як у цеха та додається у чергу на транспортування. Після цього алгоритм повертається до самого початку, на місце перевірки черги деталей на обробку.

### **Loading Truck (Вантажівка)**



*Рисунок 3.34 – 3D-модель вантажівки, що транспортує деталі між цехами*

Loading Truck – це машина, яка відповідає за генерацію деталей та їх перше розвезення серед потрібних цехів (рис. 3.34). Коли запускається симуляція, в першу чергу заповнює всі посилання місця для деталей у кузові статусом «Вільне». Після цього, запускається цикл для кожного місця в кузові, в тілі якого перевіряється, чи місце вільне, якщо відповідь позитивна, то на це місце генерується деталь з загальної бази даних деталей. Заповнивши кузов, алгоритм звертається до загальної черги складів що потребують завантаження по індексу. Якщо вказаний індекс невалідний, вантажівка очікує 1 секунду, після чого скидає значення індекса на 0 і знову повторює перевірку. Якщо елемент по індексу валідний, то запускається цикл ForEach для кожної деталі у кузові, в якому перевіряється, чи є в обраному цеху потрібні цій деталі станки. Якщо так, то цикл переривається командою «Break». Якщо після завершення циклу виявилось, що вантажівка не має потрібних для цеха деталей а кузов повністю

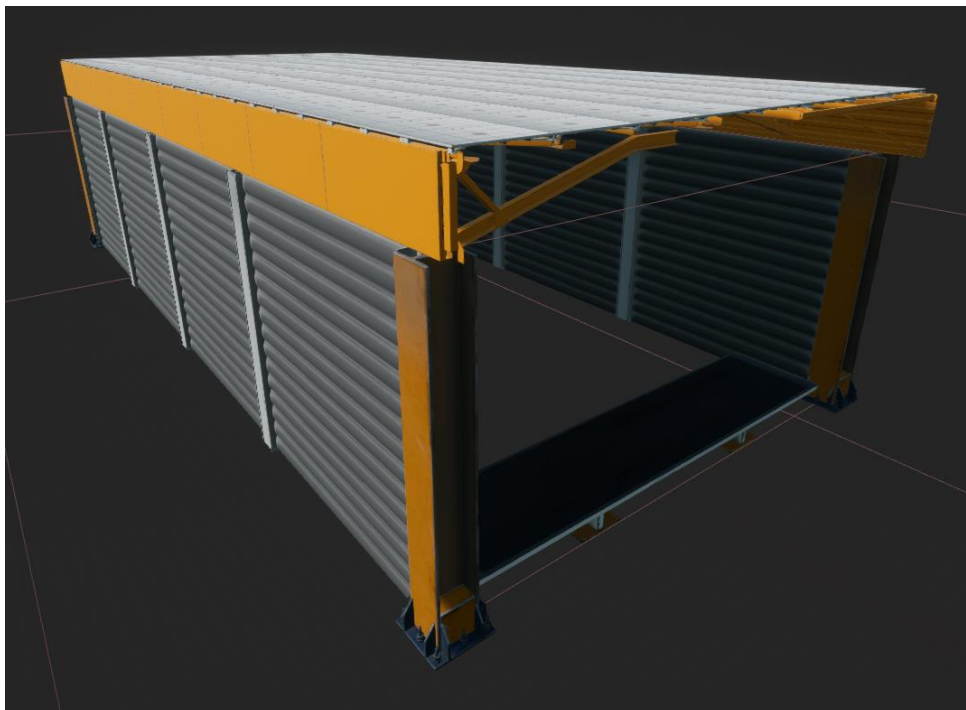
заповнений, то індекс інкрементується та алгоритм повертається до нової перевірки. Таким чином, вантажівка буде перебирати всі цехи, доки не знайде цех з потрібними станками. Якщо в кузові виявились деталі що потребують обробки у цьому цеху, то індекс обнулюється а посилання на цех встановлюється у змінну «Temp Target». Алгоритм отримує координати точки завантаження цеху та спрямовує вантажівку до неї. Коли машина досягає цілі, встановлюється кут повороту такий як у цеха та запускається цикл ForEach для кожної деталі у кузові, в ході якого отримується наступна потрібна деталі операція та перевіряється, чи є в цьому цеху потрібні станки. Якщо відповідь позитивна, то запускається ще один внутрішній цикл ForEach для кожного місця для деталей у цеху, в ході якого перевіряється, чи вільне обране місце. Якщо місце вільне, воно позначається зайнятим, а посилання на деталь додається до масива деталей на розвантаження. Цикл переривається командою «Break». Після завершення циклів та запису всіх деталей, що потрібно розвантажити до спеціального словника, в якому містяться посилання на деталі та місце, куди їх треба вивантажити, запускається цикл ForEach для кожного елемента словника, в тілі якого модель деталі відв'язується від машини, переміщується на заплановане місце, додається в чергу на обробку у цеху та видаляються зі списку деталей у кузові машини. Після завершення циклу, цех видаляється з черги на завантаження. Якщо після проведеної операції у кузові машини виявляється менше 3-х деталей, вона прямує до точки завантаження центрального складу. Після досягнення цілі, встановлюється кут повороту такий як у складу та запускається цикл ForEach для кожного місця для деталей у кузові машини. В ході циклу, якщо місце виявляється вільним, за допомогою функції «Generate Detail» на це місце генерується нова деталь з бази. Після генерації встановлюється значення індексу 0 та алгоритм повертається на початок до перевірки черги на завантаження цехів. Повна блок-схема алгоритма «Loading Truck» наведена в додатку.

### **Workshop (Цех)**

Цех – це система, що містить в собі певний набір станків, які можуть виконувати операції, та один АТМ. В середовищі цех відповідає за зберігання у собі



інформації про місця складання деталей, про чергу з деталей що очікують обробки та про передачу інформації про потрібність завантаження або розвантаження (рис. 3.35).



*Рисунок 3.35 – 3D-модель цеху цеху що використовується в середовищі*

Одразу після запуску симуляції, кожен цех запускає три цикли `ForEach` для кожного компонента типу «вектор», та відповідно до тегів «`MachinePlace`», «`UnloadDetailPlace`» та «`DoneDetailPlace`» додає посилання на них у відповідні словники де позначаються статуси місць «вільне» або «зайняте». Після цього генерується один АТМ у центрі цеха, йому передається посилання на батьківський цех та запускається симуляція у АТМ.

Далі виконується покадровий алгоритм: отримується масив глобальної бази деталей з `Game Instance`, оновлюється булева змінна що позначає статус «База не має деталей для цеха». Далі запускається цикл `ForEach` для кожної операції що виконують станки у цеху. Та запускається наступний, внутрішній цикл `ForEach` для кожної деталі в глобальній базі деталей. У тілі циклу перевіряється, чи операція деталі співпадає з операцією станка. Якщо виявляється хо одне співпадіння, цикл переривається командою «`Break`» та встановлюється статус «База має деталі для цеха». Якщо після проходження циклу виявляється що база не деталі для цеха, запускається цикл `ForEach` для кожного місця для деталей в цеху. В ході циклу перевіряється, чи вільне

місце. Після закінчення циклу, якщо вільних у цеху місць виявляється більше 9, то алгоритм додає посилання на цех у чергу цехів на завантаження у Game Instance. Далі запускається наступний цикл ForEach для кожного місця для готових деталей у цеху. В ході циклу перевіряється, чи вільне місце. Після закінчення циклу, якщо зайнятих у цеху місць виявляється більше 2, то алгоритм додає посилання на цех у глобальну базу цехів що потребують розвантаження. Блок-схема алгоритму роботи Workshop наведена у додатку.

### **Storage (Центральний склад)**

Центральний склад – це центр локації, на якій розташовані всі об'єкти симуляції (рис. 3.37). Точки біля центрального складу символізують місце, де генеруються деталі з загальної бази та місце, куди привозяться готові деталі. Центральний склад має тільки одну функцію – складання деталей. Коли вантажівка «Done Details Truck» привозить готові деталі до ключової точки, вона звертається до центрального складу, та по циклу ForEach для кожної деталі у кузові викликає функцію складу «Set Detail» (рис 3.36). Кожна наступна доставлена деталь відправляється на +20 умовних одиниць далі по осі X. Потім, функція звертається до TriggerBox, що охоплює весь склад, та дістає інформацію, чи знаходиться деталь всередині нього. Якщо ні, до нових координат деталі додається 20 умовних одиниць по осі Y.



*Рисунок 3.37 – 3D-модель центрального складу*

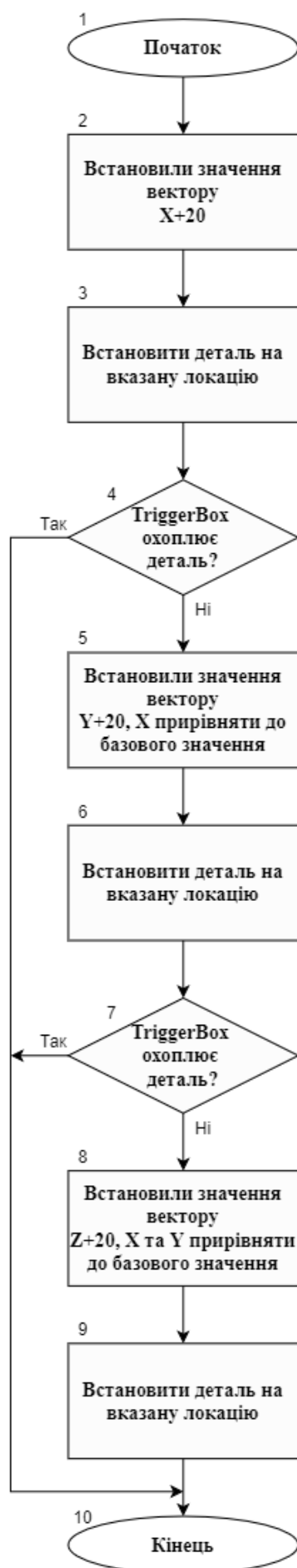
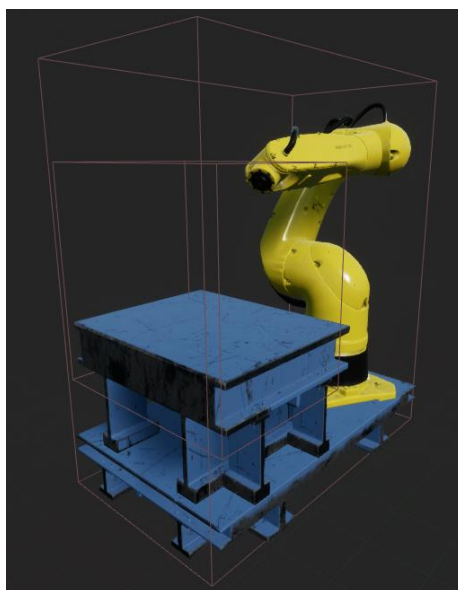


Рисунок 3.36 – Блок-схема функції «Set Detail»

Алгоритм знову звертається до TriggerBox, та дістає інформацію, чи знаходиться деталь всередині нього. Якщо ні, до нових координат деталі додається 20 умовних одиниць по осі Z. Таким чином, центральний склад може вмістити до 10000 деталей, а оскільки загальна кількість деталей обмежена 7992, то цієї кількості достатньо в повній мірі.

### **Machine (станок, ГВМ)**

Станки або ГВМ (кодова назва у середовищі «Machine» відіграють роль об'єктів, що імітують обробку деталей та демонструють анімацію з іскрами (рис. 3.39). Їх алгоритм описаний на блок-схемі (рис. 3.38), він починає свою роботу тільки коли спеціальний TriggerBox повідомляє про перехоплення об'єкта. Алгоритм отримує посилення на перехоплений об'єкт та перевіряє, чи належить об'єкт до класу «деталь». Якщо так, то станку встановлюється статус «зайнятий» та починає програватись анімація обробки деталі. Алгоритм отримує значення змінної «Delay» яка позначає час, який займає обробка цієї операції. Відбувається затримка на кількість секунд, яка вказана у значенні змінних «Delay». Після затримки у перехопленої деталі видаляється операція з індексом 0 а станок завершує програвання анімації. Деталь встановлюється на заздалегідь вказане місце для готових деталей перед станком та записується у чергу деталей що очікують обробку. Станок отримує статус «вільний».



*Рисунок 3.39 – 3D-модель станка*

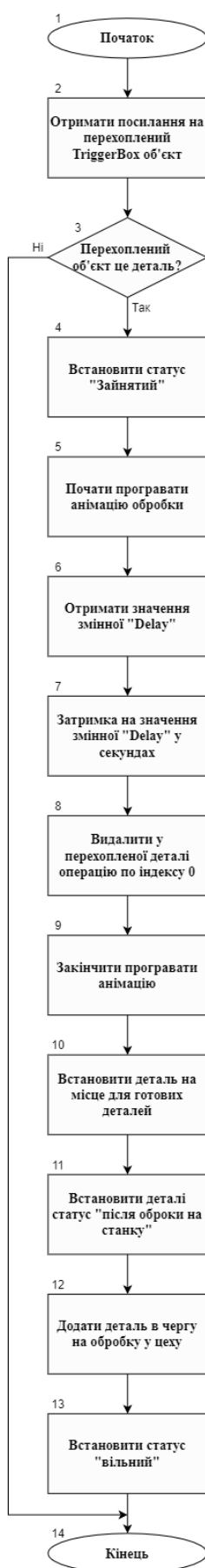


Рисунок 3.38 – Блок-схема алгоритму роботи станків

### 3.3 Система інформаційних потоків та взаємодій об'єктів процесу

Робота програми починається з запуску рівня «Main Menu» та встановлення віджета головного меню. Головне меню отримує від Save and Load інформацію про запущені та раніше збережені симуляції для ComboBox. Далі користувач може обрати 3 опції або натиснути кнопку вихода з програми. Delete Slot Menu запускається з меню та отримує від Save and Load інформацію про існуючі слоти збереження файлів. Default Level запускається з Main Menu та отримує інформацію про всі встановленні налаштування зі збереженого файла через Save and Load. Customize Menu 1 запускається з меню та отримує інформацію з Save and Load про налаштування завантаження, а також передає інформацію про свої налаштування. Customize Menu 2 запускається з Customize Menu 1 та отримує інформацію з Save and Load про налаштування завантаження, а також передає інформацію про свої налаштування. Customize Menu 3 запускається з Customize Menu 2 та отримує інформацію з Save and Load про налаштування завантаження, а також передає інформацію про свої налаштування. Customize Level запускається з Customize Menu 3 та отримує інформацію про налаштування завантаження, а також передає інформацію про свої налаштування. Customize Level також взаємодіє з Game Instance щоб заповнювати базу деталей після запуску симуляції. Game Instance зберігає в собі базу деталей, коли симуляція запущена, чергу з цехів, які потребують завантаження та чергу з цехів, які потребують розвантаження. Workshop передає або видаляє у Game Instance посилання на себе, коли потребує завантаження або розвантаження, та відповідає за генерацію АТМ і передачу йому інформації про стан черги. Також Workshop містить у собі інформацію про місця складання деталей та чергу деталей, що потребують обробки. АТМ отримує інформацію про стан черги деталей у Workshop а також може видаляти деталі з черги. АТМ також отримує інформацію про стан станків. Machines (ГВМ або станки) передають у Workshop інформацію про виконувані операції а також взаємодіють з деталями та можуть видаляти у них виконані операції. Trucks або вантажівки отримують від деталей інформацію про наступні потрібні їм операції, отримують інформацію про станки через Workshop,

взаємодіють з чергами цехів через Game Instance. Вантажівки також передають готові деталі до центрального складу (рис. 3.40).

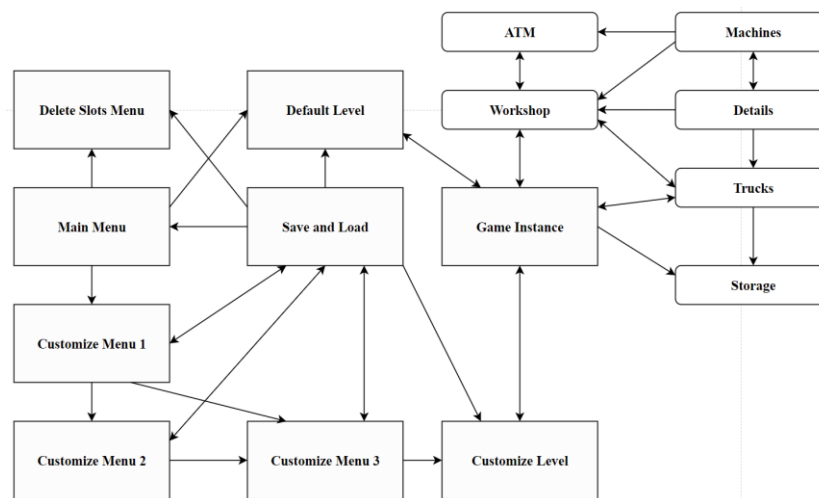


Рисунок 3.40 – Схема системи інформаційних потоків та взаємодій суб'єктів процесу

### Висновок до розділу

У розділі обґрунтовано вибір середовища розробки, детально описано архітектуру та структуру Unreal Engine 4, засоби роботи з об'єктами, розглянуто технології роботи з 3D-графікою.

Детально описано архітектуру та алгоритми розробленого проекту: функціональність кнопок головного та додаткових меню, функції збереження та завантаження налаштувань з файлу. Наведено скріншоти роботи з меню налаштувань в додатку, детально описано кожну стадію та наведено блок-схеми алгоритмів їх роботи. Окремо описано об'єкти середовища, які беруть участь у моделюванні та симуляції в реальному часі:

- Деталь – об'єкт у середовищі, який зберігає набір операцій що треба пройти: описано алгоритм роботи, наведено скріншот 3D-моделі та блок-схему.
- АТМ – відповідає за транспортування деталей в цеху відповідно черзі: наведено скріншот 3D-моделі та блок-схему у додатку.
- Вантажівка – відповідає за генерацію деталей та їх перше розвезення серед потрібних цехів: наведено скріншот 3D-моделі та блок-схему у додатку.

- Цех – відповідає за зберігання у собі інформації про місця складання деталей, про чергу з деталей що очікують обробки та про передачу інформації про потрібність завантаження або розвантаження: наведено скріншот 3D-моделі та блок-схему у додатку.
- Центральний склад – відповідає за оптимальне складання деталей: наведено скріншот 3D-моделі та блок-схему роботи алгоритму.
- Станок – відіграє роль об'єкта, що імітує обробку деталей та демонструє анімацію: наведено скріншот 3D-моделі та блок-схему роботи алгоритму.

Розроблений проект знаходиться в робочому стані та відповідає поставленим вимогам.



## РОЗДІЛ 4. МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП-ПРОЕКТУ

### 4.1. Опис ідеї проекту

Таблиця 4.1. Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Розробка гнучкого програмного забезпечення, за допомогою якого технолог зможе на основі самостійно введених налаштувань побудувати систему гнучкого виробництва та переглянути симуляцію його роботи в реальному часі в 3D-просторі.	Застосування технологом виробництва для швидкого моделювання нових ситуацій щоб уникнути простоїв обладнання та побачити слабкі місця системи.	<ul style="list-style-type: none"> <li>– Швидке реагування на команди користувача;</li> <li>– Наявність функцій що упереджають випадкові дії зі сторони користувача;</li> <li>– Наявність функцій що дозволяють моделювати виробництво у 3D-середовищі з якісною графічною системою;</li> <li>– Наявність функцій збереження та завантаження налаштувань;</li> <li>– Легкість освоєння і застосування;</li> <li>– Баланс між простотою і доступністю функцій і даних;</li> </ul>
	Застосування при проектуванні виробництва, щоб заздалегідь передбачити недоліки системи, слабкі місця та неоптимальну структуру.	

Таблиця 4.2. Опис ідеї стартап-проекту

№	Техніко-економічні характеристики ідеї	Продукція конкурентів			W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проєкт	DELMIA	AnyLogic			
1	Швидкість роботи	В	С	Н	Потрібний час на завантаження вікон	Зрозуміла архітектура	Легкість навчання
2	Вимоги до системи	В	С	С	Якісна 3D-графіка	Інтерфейс	Оптимізація
3	Зручність використання	В	С	В	Потрібне розуміння архітектури гнучких виробництв	Наявність функцій збереження та завантаження налаштувань	Легкість освоєння і застосування

## Закінчення таблиці 4.2.

5	Якість 3D-графіки	В	Н	Н	Спрощені анімації	3D-об'єкти з безкоштовної бази	Якісні 3D-моделі, текстури та освітлення сцени
6	Точність моделювання	С	В	В	Відсутність функцій зміни розмірів об'єктів	Широкі можливості для позиціювання	Якісні колізії 3D-об'єктів

**4.2. Технологічний аудит ідеї проєкту**

Таблиця 4.3. Технологічна здійсненність ідеї проєкту

№	Ідея проєкту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Моделювання виробництва в цілях виявлення слабких місць	Розробка додатку на Unreal Engine 4 з упором на точність моделювання	Використання якісних 3D моделей та проробка точних колізій між об'єктами	У вільному та безкоштовному доступі
2	Моделювання виробництва в цілях проектування	Розробка додатку на Unreal Engine 4 з упором на широкий функціонал зміни положення та розмірів об'єктів	Використання безкоштовних або платних плагінів з Unreal Marketplace для роботи з проектуванням та розташуванням об'єктів	У вільному та безкоштовному доступі
3	Моделювання виробництва в цілях розрахунку рентабельності	Розробка додатку на Unreal Engine 4 з упором на статистичні дані та інфографіки	Використання безкоштовних або платних плагінів з Unreal Marketplace для роботи з статистикою та інфографіками	У вільному та безкоштовному доступі
<i>Обрана технологія реалізації ідеї проєкту: 1</i>				

Висновок: можлива технологічна реалізація продукту, обрана технологія №1.

### 4.3. Аналіз ринкових можливостей запуску стартап-проєкту

Таблиця 4.4. Попередня характеристика потенційного ринку

№	Показники стану ринку	Характеристика
1	Кількість головних гравців, од	3
2	Загальний обсяг продаж, грн./ум.од	3500
3	Динаміка ринку	Ринок гнучких виробництв та впровадження автоматизації зростає щороку
4	Наявність обмежень для входу	Немає
5	Специфічні вимоги до стандартизації та сертифікації	Немає
6	Середня норма рентабельності в галузі або по ринку, %	80%

Висновок: враховуючи кількість головних гравців по ринку, зростаючу динаміку ринку, невелику кількість конкурентів та середню норму рентабельності можна зробити висновок, що на даний момент, ринок для входження є привабливим.

Таблиця 4.5. Характеристика потенційних клієнтів стартап-проєкту

№	Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці цільових груп клієнтів	Вимоги споживачів до товару
1	Передбачення виникнення помилок, швидке адаптування до виробництва нових деталей або нових умов роботи підприємства	Технологи на гнучких підприємствах	Швидкість роботи з додатком, прагнення отримання результату якнайшвидше	Наявність знань про технічні складові процесу роботи гнучких підприємств
2	Передбачення схеми оптимальної роботи виробництва, передбачення помилок, слабких місць	Спеціалісти що відповідають за проектування структури гнучких підприємств	Приділення уваги зручному місцерозташуванню об'єктів виробництва, прагнення передбачити проблемні місця без тестування на реальних об'єктах	
3	Розрахунок рентабельності підприємства	Маркетологи гнучких підприємств	Прагнення чітко передбачити ефективність виробництва	

Таблиця 4.6. Фактори загроз

№	Фактор	Зміст загрози	Можлива реакція компанії
1	Конкуренти	Наявність конкурентів які розробляють схожий додаток	Зменшення ціни на програмне забезпечення; Додавання нових, унікальних функцій програмному додатку; Надання ліцензій на обслуговування; Надання своїх спеціалістів;
2	Кошти на розробку та підтримку продукту	Закінчення виділеного бюджету та недостатнє фінансування	Участь в ІТ-конкурсах задля привертання уваги та нових інвесторів до програмного продукту; Мотивація роботи на перспективу; Розробка додатку частково, по ітераціям, щоб покроково виводити на ринок, пригортати до себе увагу потенційних інвесторів та отримувати відгуки від користувачів;
3	Вихід аналогу	Вихід аналогу програмного додатку	Випуск програми у коротші строки з неповною функціональністю для зацікавлення цільової аудиторії; Проведення рекламної компанії;

Таблиця 4.7. Фактори можливостей

№	Фактор	Зміст можливості	Можлива реакція компанії
1	Новий продукт	Вихід на ринок, Зменшення монополії, Надання нових рішень у сфері	Розробка нових функціональних можливостей; Випуск нового програмного забезпечення на ринок;
2	Вихід аналогу	Надати програмне забезпечення з новими характеристиками та можливостями що відсутні у програмі конкурентів	Аналіз ринку та цільової аудиторії щоб задовільнити їх потреби та надати функціональність у найкоротші строки за ціну, яка є дешевшою ніж у конкурентів.
3	Зворотній зв'язок від користувачів	Можливість отримання необхідної інформації для виправлення багів та вдосконалення продукту	Отримання вхідних даних допрацювання програмного забезпечення задля задовільнення потреб кінцевих користувачів.
4	Грошова винагорода за рекламу	При достатньому попиту на програмний додаток можлива комерціалізація продукту на основі реклами	Введення реклами; Точкова комерціалізація продукту; Влиття додаткових коштів у проект щоб забезпечити його подальший розвиток.

Таблиця 4.8. Ступеневий аналіз конкуренції на ринку

№	Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1	Тип конкуренції: монополістична	Товар від кожної компанії на ринку, являється недосконалим заміником товару, реалізованого іншими фірмами; На ринку є умови для входу та виходу; Ціна корелює між суперниками;	Розробка продукту з характеристиками, які покривають сфери застосування що не покривають інші додатки; Кореляція цін у відповідності до інших додатків;
2	Рівень конкурентної боротьби: світовий	Всі продукти замітники розроблялись інтернаціональними командами з різних куточків світу, продукти не належать до певної держави, а належать команді розробників	Вихід на ринок збуту продукту з клієнто-необхідною функціональністю; Налагодження маркетингу на основних інтернет-ресурсах задля охоплення великої кількості потенційних користувачів; Надання бета-версій продукту.
3	Галузева ознака: внутрішньогалузева	Даний тип продукту може використовуватися тільки у сфері розробки ІТ додатків \ продуктів	Надання зручного, інтуїтивно зрозумілого інтерфейсу; Підтримка всім відомих методів взаємодії з середовищем розробки; Наявність документації та он-лайн підтримки.
4	Конкуренція за видами товарів: товарно-видова	Дана конкуренція – конкуренція між товарами одного виду.	Впровадження функціональності яка відсутня у інших додатків; Спрощення інтерфейсів; Надання підтримки.
5	Характер конкурентних переваг: цінова та не цінова	Цінові переваги – точкова комерціалізація; Не цінова – надання функціональності, що відсутня у товарах-замінниках.	Надання платних ліцензій лише на критично важливу функціональність для клієнта з певним строком підтримки, що зазначена у відповідній ліцензії; Впровадження унікальної функціональності.
6	За інтенсивністю: марочна	Наявність унікального знаку що відрізняє даний продукт від продуктів-замінників	Впровадження власної назви та власного знаку.

Таблиця 4.9. Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Anylogic, DELMIA	Відсутні	The AnyLogic Company, Dassault Systems	Онлайн-магазини що продають додатки	Програмні додатки що підтримують 3D-моделювання та написання алгоритмів
Висновки	Розширюють свою діяльність, концентруються на різних напрямках програмних продуктів	Відсутні	Постачальники контролюють розробку програмних продуктів, відповідають за рекламу.	Клієнти диктують ринкові умови, надають розробникам відгуки та нові цілі для покращення додатків	Можливість покращення програмного продукту

Проаналізувавши можливості роботи на ринку з огляду на конкурентну ситуацію можна зробити висновок: оскільки кожний з існуючих продуктів не впливає у великій мірі на поточну ситуацію на ринку в цілому, кожний з існуючих продуктів має свою специфічну сферу використання та свої позитивні та негативні сторони щодо рішення певних типів задач, то робота та вихід на даний ринок є можливою і реалізованою задачею.

Для виходу на ринок продукт повинен мати функціонал що відсутній у продуктів-аналогів, повинен задовольняти потреби користувачів, мати необхідний та достатній функціонал з конфігурування, підтримку зі сторони розробників та можливість розробки спеціального функціоналу за відповідною ліцензією.

Таблиця 4.10. Обґрунтування факторів конкурентоспроможності

№	Фактор конкурентоспроможності	Обґрунтування
1	Швидке реагування на команди користувача.	Швидкість та зручність роботи з програмою – потужна перевага будь-якого конкурентноспроможного додатку.

Закінчення таблиці 4.10.

2	Наявність функцій що упереджають випадкові дії зі сторони користувача.	Перед кожною важливою дією, результат якої виправити буде неможливо або це займе більше часу, додаток від користувача буде очікувати додаткового підтвердження
3	Наявність функцій що дозволяють моделювати виробництво у 3D-середовищі з якісною графічною системою.	Основна задача додатку – відобразити якісну симуляцію виробництва зі всіма можливими подробицями у 3D-просторі.
4	Доступність функцій маніпулювання об'єктами в середовищі додатка.	Користувач може самостійно розташовувати цехи в середині середовища, змінювати їх положення та кут повороту.
5	Наявність функцій збереження та завантаження налаштувань.	Щоб пришвидшити роботу з програмою та випадково не втратити введені налаштування в додатку доступні функції збереження вікон інтерфейса та інших налаштувань.
6	Легкість освоєння і застосування.	Інтуїтивно зрозумілий інтерфейс, приємні кольори та проста архітектура програми дозволить будь-якому користувачу з легкістю освоїти додаток.
7	Однаковість методів роботи з системою, однаковість стилів та кольорів інтерфейсу користувача.	Простота в освоєнні програми, частини інтерфейсу що запам'ятовуються інтуїтивно – дуже важлива частина дружнього інтерфейсу користувача.
8	Оптимізація.	Швидкість відгуку програми, швидке виведення результатів та моментальне відображення якісної 3D-графіки привабить користувача.

Таблиця 4.11. Порівняльний аналіз сильних та слабких сторін додатку

№	Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів у порівнянні з запропонованим						
			-3	-2	-1	0	+1	+2	+3
1	Швидке реагування на команди користувача.	3			AL	DM			
2	Наявність функцій що упереджають випадкові дії зі сторони користувача.	5				AL	DM		
3	Наявність функцій що дозволяють моделювати виробництво у 3D-середовищі з якісною графічною системою.	10	AM	DM					

Закінчення таблиці 4.11.

4	Доступність функцій маніпулювання об'єктами в середовищі додатка.	4		DM	AL				
5	Наявність функцій збереження та завантаження налаштувань.	0		AL				DM	
6	Легкість освоєння і застосування.	9	DM	AL					
7	Однаковість методів роботи з системою, однаковість стилів та кольорів інтерфейсу користувача.	0				AL	DM		
8	Оптимізація.	8	DM			AL			

Таблиця 4.12. SWOT аналіз стартап-проєкту

<p>Сильні сторони (S):</p> <p>Швидке реагування на команди користувача</p> <p>Функції що дозволяють моделювати виробництво у 3D-середовищі з якісною графічною системою</p> <p>Легкість освоєння і застосування</p>	<p>Слабкі сторони (W):</p> <p>Вузьконаправленність</p> <p>Можливе виникнення системних помилок</p> <p>Не може працювати на слабких системах</p>
<p>Можливості (O):</p> <p>Приватні контракти</p> <p>Державні контракти</p> <p>Постійна підтримка та розширення функціональності додатку</p>	<p>Загрози (T):</p> <p>Складність передбачення роботи алгоритмів при непередбачуваних ситуаціях</p> <p>Потужні конкуренти</p>

Таблиця 4.13. Альтернативи ринкового впровадження стартап-проєкту

№	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Безкоштовне надання певного функціоналу у користування споживачам на обмежений термін	Головний ресурс – люди, даний ресурс - наявний	1-3 місяці
2	Реклама	Залучення власних коштів для реклами товару	1 місяць
3	Написання статей та опис товару на відомих ресурсах	Головний ресурс – час, даний ресурс - наявний	2-3 тижні
4	Презентація товару на хакатонах й інших ІТ заходах	Ресурс – час та гроші для участі, наявні	1-2 місяці



#### 4.4. Розроблення ринкової стратегії проєкту

Таблиця 4.14. Вибір цільових груп потенційних споживачів

№	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Технологи на гнучких підприємствах	Присутня	Високий	Середня	Легка
2	Спеціалісти що відповідають за проектування структури гнучких підприємств	Присутня	Високий	Висока	Середня
3	Маркетологи гнучких підприємств	Присутня	Низький	Висока	Середня
Які цільові групи обрано: 1, 2, 3					

Отже, відповідно до проведеного аналізу, основною цільовою аудиторією для розповсюдження додатку є працівники гнучких підприємств, які відповідають за безперервну обробку жеталей, проектування підприємств та маркетологи цих підприємств. Обрано стратегію таргетингового маркетингу, оскільки програмне забезпечення підходить тільки для вузьконаправлених підприємств та працівників. Клієнтам буде надаватися пробна версія продукту, повну версію програми можна буде придбати з можливістю розширення функціональності за домовленістю.

Таблиця 4.15. Визначення базової стратегії розвитку

Обрана альтернатива розвитку проєкту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
Надання унікальної вузьконаправленої функціональності яка відсутня у конкурентів, надання свого спеціаліста	Участь у конкурсах, конференціях, проведення рекламної компанії.	Можливість швидко додати потрібні для користувачів функції на основі їх відгуків; Якісна та оптимізована 3D-графіка; Відмітні характеристики товару;	Стратегія диференціації

Таблиця 4.16. Визначення базової стратегії конкурентної поведінки

Чи є проєкт «першопрохідцем» на ринку	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, які?	Стратегія конкурентної поведінки
Ні, оскільки є схожі додатки, але вони не мають повного функціоналу	Так, ціль компанії знайти нових клієнтів та забрати вже існуючих у конкурентів задля збільшення продажів	Основна ціль - це розробка нового унікального функціоналу а також запозичення важливих та потрібних функцій у додатків-конкурентів	Стратегія лідера

Таблиця 4.17. Визначення стратегії позиціонування

№	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проєкту	Вибір асоціацій, які мають сформувати комплексну позицію власного проєкту
1	Ефективність	Оновлення та розширення	Перспективи розширення функціоналу та розвиток проєкту	Швидке освоєння
2	Точність моделювання	Оновлення та розширення	Легке освоєння та доступність функцій збереження налаштувань	Якісна робота алгоритмів
3	Наглядна симуляція	Заняття місця конкурентів	Швидке реагування на дії користувача, оптимізація коду та графіки	Якісна 3D-графіка

Отже, можна зробити висновок, що стартап-компанія обирає як базову стратегію розвитку – стратегію оновлення та розширення, як базову стратегію конкурентної поведінки – стратегію заняття місця конкурентів.

#### 4.5. Розроблення маркетингової програми стартап-проєкту

Таблиця 4.18. Визначення ключових переваг концепції потенційного товару

№	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	На виробництві	Швидкість роботи з додатком, швидке отримання потрібного результату	Додаток направлений на моделювання вузьконаправлених підприємств, тому в ньому заздалегідь встановлені певні налаштування та функції для зручності
2	При проектуванні	Попереднє передбачення недоліків системи, слабких та неоптимальних місць, визначення об'ємів виробництва	Робота з додатком направлена на роботу з вузьким колом можливих виробництв, тому змодельовавши систему проектувальник одразу може побачити більшість недоліків

Таблиця 4.19. Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
1. Товар за задумом	Система моделювання гнучкого виробництва		
2. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх/Тл/Е/Ор
	Швидкість роботи	Нм	Тл
	Легкість освоєння	Нм	Тл
	Оптимізація	Нм	Тх
	Якість 3D-графіки	Нм	Е
	Зручність	Нм	Ор
	Точність моделювання	Нм	Тх
	Дружній інтерфейс	Нм	Е
	Додаток поставляється у вигляді .exe-файлу з інсталятором		
	Марка: RealPlan		
3. Товар із підкріпленням	До продажу: наявні промокоди на придбання декількох ліцензій, знижки на покупку товару, повна документація.		
	Після продажу: додаткова підтримка з боку розробників, надання своїх спеціалістів для допомоги в налаштуванні		
За рахунок чого потенційний товар буде захищено від копіювання: патент, захист інтелектуальної власності.			

Таблиця 4.21. Визначення меж встановлення ціни

Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
500-2000\$	100-2000\$	1000-1000000\$/місяць	60-200\$

Таблиця 4.22. Формування системи збуту

Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
Клієнти зможуть придбати додаток на основі ліцензії	Забезпечити можливість придбати додаток кожному бажаному	Розробник – посередник - клієнт	Роздрібна

Таблиця 4.23. Концепція маркетингових комунікацій

№	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	Технологи	Інтернет-форуми, соціальні мережі, конференції, приватні зустрічі	Швидкість моделювання	Демонстрація широкого функціоналу додатку	Рекламне звернення повинно бути націлено на інтереси потенційних клієнтів, статистику, інфографіки та якість
2	Проектувальники		Точність моделювання	Демонстрація якості виконання додатку	
3	Маркетологи		Виведення статистики, економія	Демонстрація статистичних таблиць в результаті роботи з додатком	

Як результат було створено маркетингову програму, що включає в себе визначення ключових переваг концепції потенційного товару, опис моделі товару, визначення меж встановлення ціни, формування системи збуту та концепцію маркетингових комунікацій.

### **Висновки по розділу**

В розділі детально описано маркетинговий аналіз стартап-проекту. А саме: описано ідею проекту, розглянуто технологічний аудит ідеї проекту, технологічну здійсненність ідеї проекту, проведено аналіз ринкових можливостей – дано попередню характеристику потенційного ринку та характеристику потенційних клієнтів проекту. Описано фактори загроз, фактори можливостей, зроблено ступеневий аналіз конкуренції на ринку, аналіз конкуренції в галузі за М. Портером. Обґрунтовано фактори конкурентоспроможності, зроблено порівняльний аналіз сильних та слабких сторін додатку. Проведено SWOT-аналіз проекту, визначено альтернативи ринкового впровадження. Розроблено ринкову стратегію проекту: обрано цільові групи потенційних споживачів, визначено базову стратегію розвитку, базову стратегію конкурентної поведінки, визначено стратегію позиціонування. Розроблено маркетингову програму стартап-проекту: визначено ключові переваги концепції потенційного товару, описано три рівні моделі товару, визначено межі встановлення ціни та зформовано систему збуту. Описано концепцію маркетингових комунікацій.

## ВИСНОВКИ

В рамках магістерської дисертації вивчено предметну область, визначено предмет та об'єкт дослідження, розглянуто актуальність та описано новизну програмного продукту. Наведено цифри очікуваного зростання ефективності у виробничих галузях завдяки автоматизації та штучному інтелекту. Проаналізовано два існуючі рішення – програмі додатки Anylogic та DELMIA, описано їх основний функціонал та недоліки. На основі отриманої інформації поставлено задачу дисертації та визначено вимоги до програмного забезпечення, що буде розроблено, враховуючи недоліки вже існуючих рішень.

Розглянуто актуальність впровадження автоматизації на виробництвах, це дає змогу забезпечити швидкий доступ до даних по усім автоматизованим частинам виробництва та постійно контролювати показники діяльності та ефективності компанії, результатом роботи якої є ефективне функціонування об'єкта. Описано підвищення ефективності при впровадженні автоматизації у виробництво, розглянуто фактори, що впливають на ефективність АСУ. Розглянуто об'єктно-орієнтований підхід до проектування автоматизованої системи керування виробництвом та проаналізовано структуру автоматизованих систем, їх взаємодію зі штучним інтелектом та структуру гнучких виробничих систем.

Обґрунтовано вибір середовища розробки, детально розглянуто архітектуру та структуру Unreal Engine 4, засоби роботи з об'єктами, розглянуто технології роботи з 3D-графікою. Описано архітектуру та алгоритми розробленого проекту: функціональність кнопок головного та додаткових меню, функції збереження та завантаження налаштувань з файлу, маніпулювання об'єктами та процес роботи симуляцій. Наведено скріншоти роботи додатку на різних стадіях моделювання, детально описано кожен етап та наведено блок-схеми алгоритмів роботи. Описано об'єкти середовища, які беруть участь у моделюванні, їх структуру та систему взаємодії. Розроблений проект детально протестовано. Додаток знаходиться в робочому стані та відповідає поставленим у вимогам.

Описано маркетинговий аналіз стартап-проекту – ідею проекту, розглянуто технологічний аудит ідеї проекту, технологічну здійсненність ідеї проекту, проведено

аналіз ринкових можливостей – дано попередню характеристику потенційного ринку та характеристику потенційних клієнтів проекту. Описано фактори загроз, фактори можливостей, зроблено ступеневий аналіз конкуренції на ринку, проведено аналіз конкуренції в галузі за М. Портером. Обґрунтовано фактори конкурентоспроможності, зроблено порівняльний аналіз сильних та слабких сторін додатку. Проведено SWOT-аналіз проекту, визначено альтернативи ринкового впровадження. Розроблено ринкову стратегію проекту: визначено базову стратегію розвитку, проаналізовано цільові групи потенційних споживачів, базову стратегію конкурентної поведінки, визначено стратегію позиціонування. Розроблено маркетингову програму стартап-проекту: визначено ключові переваги концепції потенційного товару, описано три рівні моделі товару, визначено межі встановлення ціни та зформовано систему збуту. Описано концепцію маркетингових комунікацій.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Реалізація моделювання складної просторової моделі з відображенням роботи штучного інтелекту [Електронний ресурс] / І. М. Пінчук, А. Б. Пластовець, Л. С. Ямпольський // Адаптивні системи автоматичного управління. - 2010. - № 17. - С. 94-98. - Режим доступу: [http://nbuv.gov.ua/UJRN/asau\\_2010\\_17\\_14](http://nbuv.gov.ua/UJRN/asau_2010_17_14)
2. Гибкие робототехнические системы / Гавриш А. П., Ямпольский Л. С.. // Київ, Головне видавництво видавничого об'єднання «Вища школа». – 1989.
3. Мультиагентне середовище поетапного синтезу системи динамічного керування у гнучкій виробничій системі / С. О. Дьяков, Л. С. Ямпольський // Адаптивні системи автоматичного управління № 1' (28) 2016
4. Тіщенко Н.М. Автоматизоване проектування систем автоматизованого проектування. – М: Энергоиздат. 1986. – 334 с.
5. Бесекерський В.А. та ін. Руководство по проектированию систем автоматизованого керування. – М: Высшая школа. 1983. – 296 с.
6. Гибкие производственные системы. / Меткін Н. П., Лапін М. С., Клейменов С. А., Крітський В. М.. – 1989.
7. Балашов А.І., Виробничий менеджмент на підприємстві. – Санкт-Петербург: Питер, 2009. - 160 с.
8. Голоктеев К., Матвеев І. Керування виробництвом: інструменти, які працюють. - Санкт-Петербург: Питер, 2008. - 251 с.
9. Оцінка прийняття рішень при виборі промислових роботів гнучкої виробничої системи [Електронний ресурс] / О. І. Лісовиченко, Є. С. Пуховський, Л. С. Ямпольський // Адаптивні системи автоматичного управління. - 2006. - № 9. - С. 126-132. - Режим доступу: [http://nbuv.gov.ua/UJRN/asau\\_2006\\_9\\_17](http://nbuv.gov.ua/UJRN/asau_2006_9_17)
10. Алгоритм синтезу моделі ГВС в об'єктно-орієнтованому середовищі моделювання / О. І. Лісовиченко, Л. С. Ямпольський, О. А. Лавров, Є. С. Пуховський // Адаптивні системи автоматичного управління. - 2007. - № 10. - С. 69-83. - Режим доступу: [http://nbuv.gov.ua/UJRN/asau\\_2007\\_10\\_11](http://nbuv.gov.ua/UJRN/asau_2007_10_11)
11. Динамічне планування у виробничих системах в умовах невизначеності [Електронний ресурс] / С. О. Дьяков, Л.



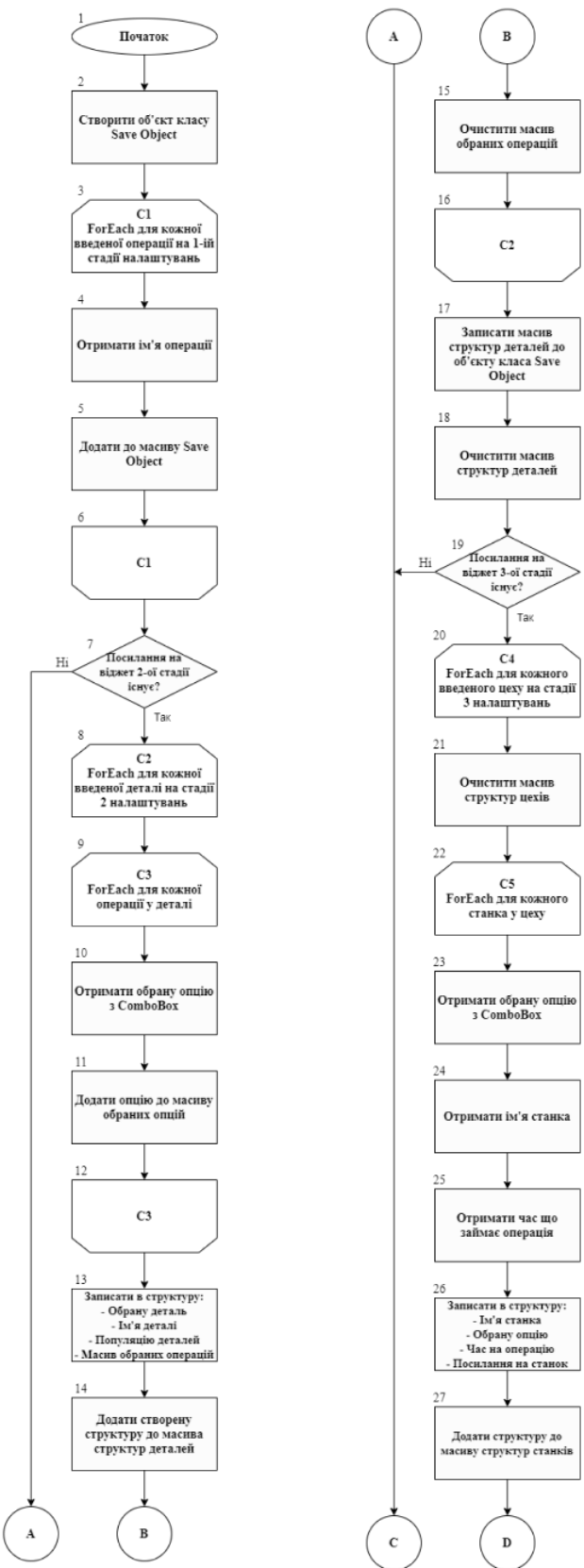
- С. Ямпольський // Технологічні комплекси. - 2014. - № 2. - С. 22-26. - Режим доступу: [http://nbuv.gov.ua/UJRN/Tehkom\\_2014\\_2\\_5](http://nbuv.gov.ua/UJRN/Tehkom_2014_2_5)
12. A future that works: automation, employment, and productivity / McKinsey & Company / January 2017. – Режим доступу: <https://www.mckinsey.com/~media/mckinsey/featured%20insights/Digital%20Disruption/Harnessing%20automation%20for%20a%20future%20that%20works/MGI-A-future-that-works-Executive-summary.ashx>
13. Artificial Intelligence Strategy / The Federal Government // November 2018. – Режим доступу: <https://www.ki-strategie-deutschland.de/home.html>
14. The Benefits of Factory and Manufacturing Automation for Today's Worker / Режим доступу: <https://www.smartsheet.com/factory-manufacturing-automation>
15. Документація Unreal Engine 4 / Режим доступу: <https://docs.unrealengine.com/en-US/index.html>
16. Unreal Engine 4 / Режим доступу: <https://www.unrealengine.com/en-US/>
17. DELMIA / Режим доступу: <https://www.3ds.com/ru/produkty-i-uslugi/delmia/>
18. AnyLogic / Режим доступу: <https://www.anylogic.ru>
19. Підхід до створення системи моделювання гнучкого виробництва / А. Shcherbinsky, О. Lisovychenko, О. Bondarenko. // Адаптивні системи автоматичного управління №35. – 2019. – с. 88–97.

## ДОДАТКИ

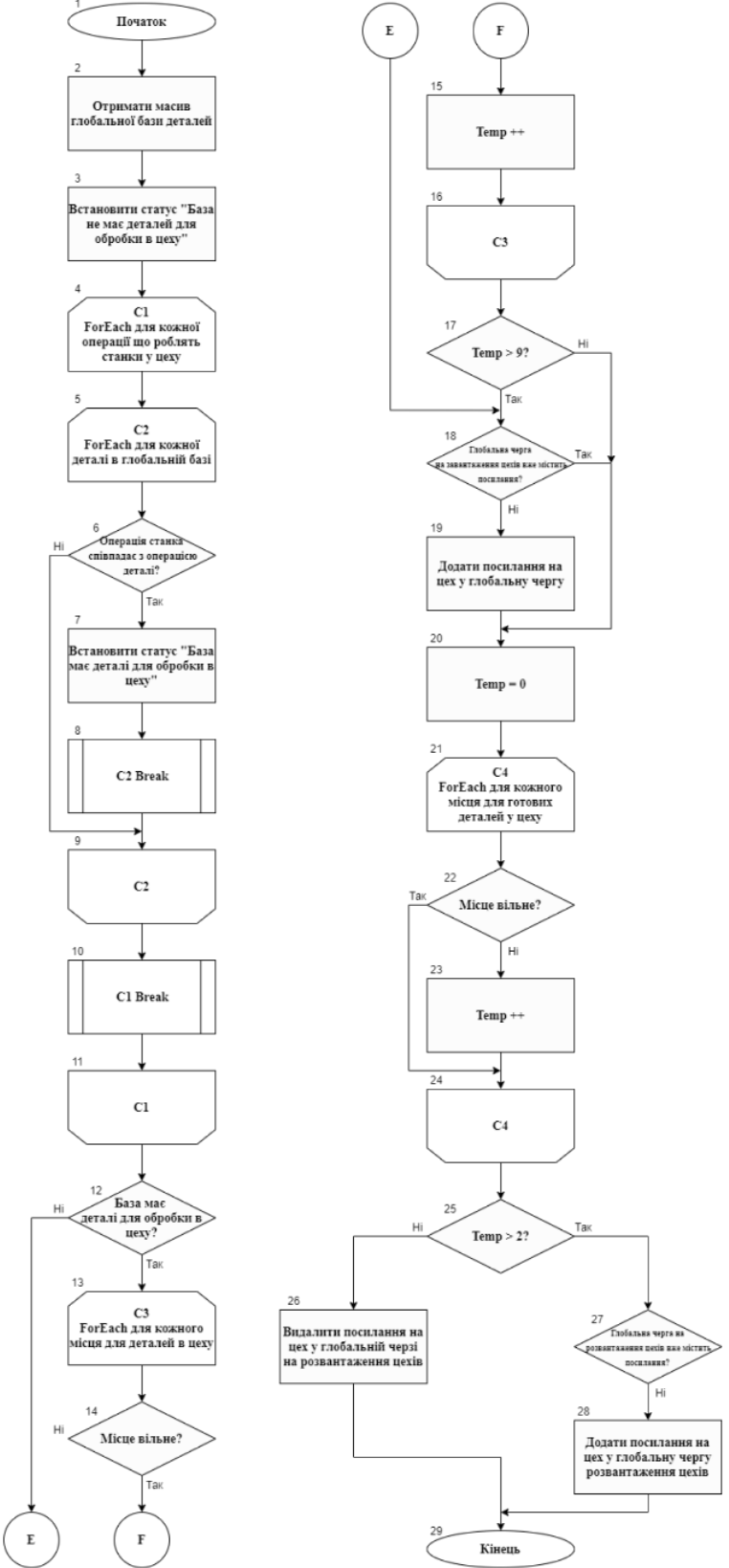
## ДОДАТОК А

Блок-схеми алгоритмів «Save» та «Workshop»

Save



Workshop



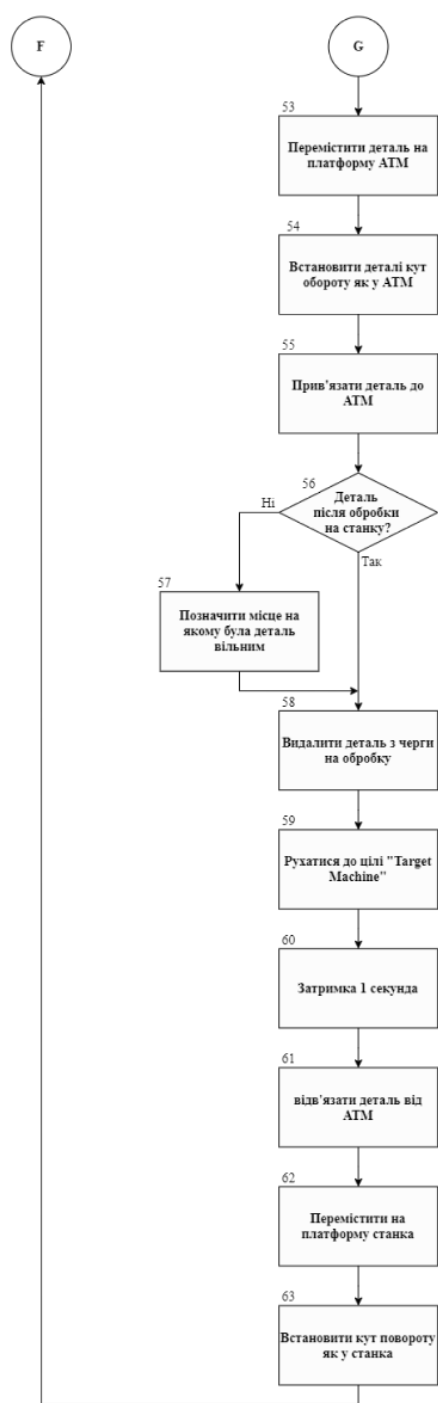
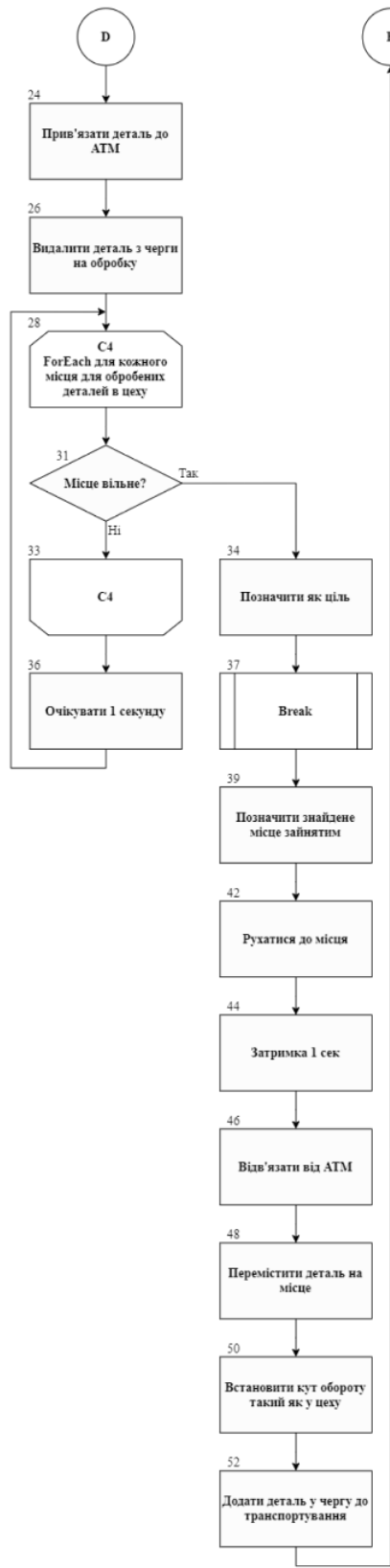
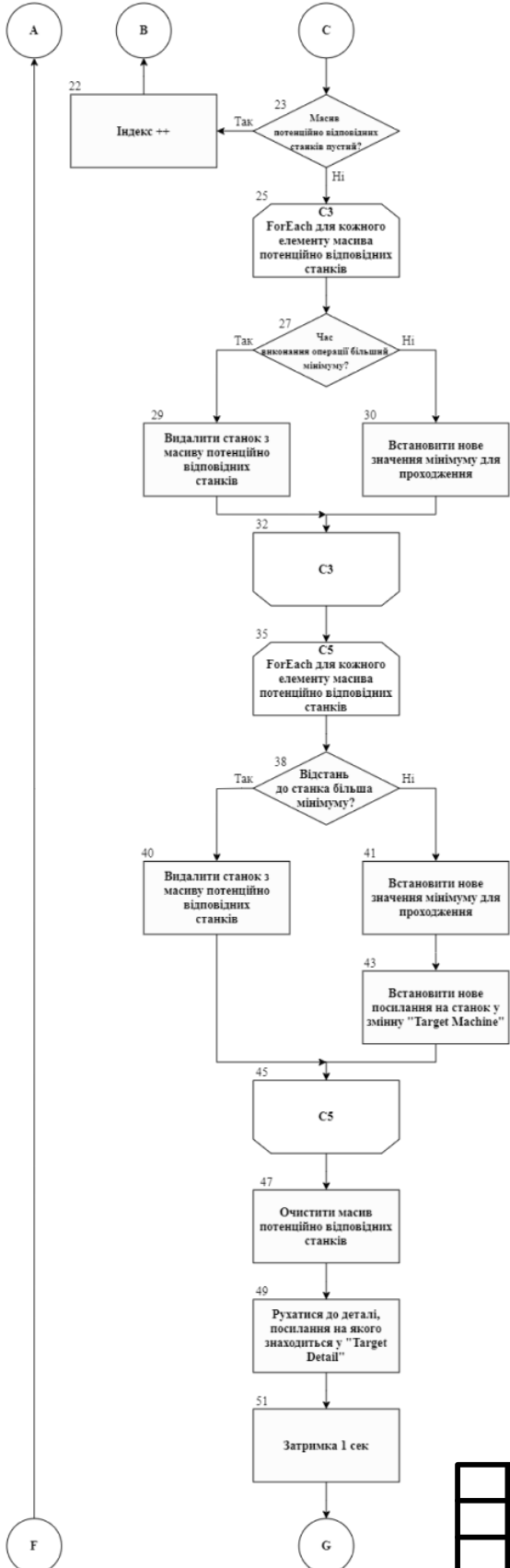
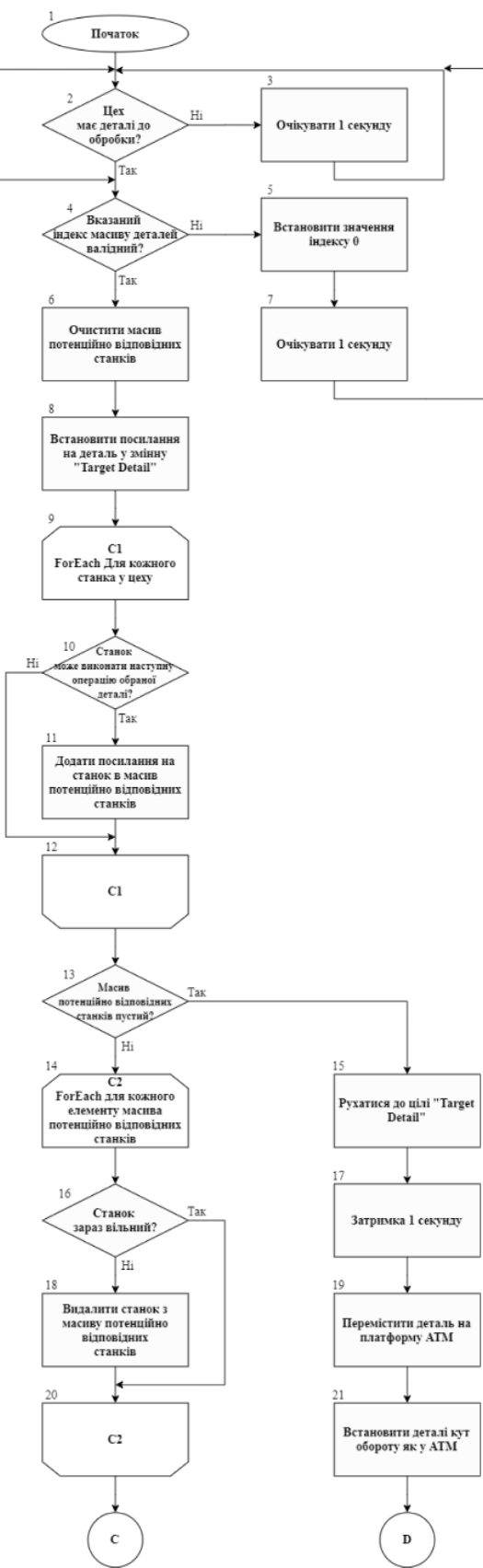
Підпис і дата	Інв. № дубл.	Взам. інв. №	Підпис і дата	Інв. № ориг.

IK-91.32 3132.02				
Блок-схеми алгоритмів «Save» та «Workshop»				
Літ. Маса Мірило				
Лист Листів				
Група ІК-91мп				

Зм.	Лист	№ докум	Підпис	Дата
Розроб.	Щербінський А.О.			
Перев.	Лісовиченко О.І.			
Н. контроль	Пасько В.П.			
Затв.	Пархомей І.Р.			

## ДОДАТОК Б

### Блок-схема алгоритму «АТМ»



Блок-схема алгоритму «АТМ»

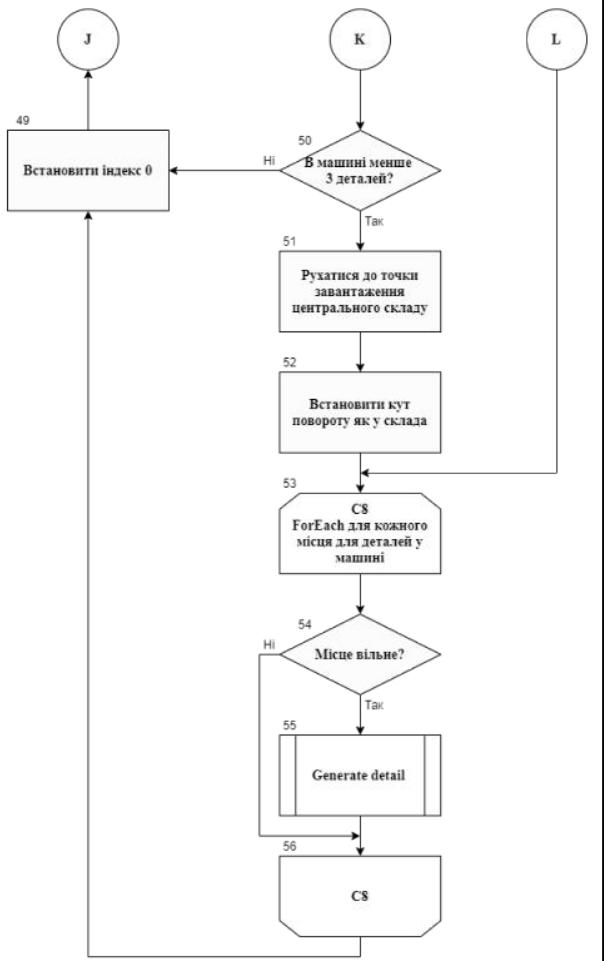
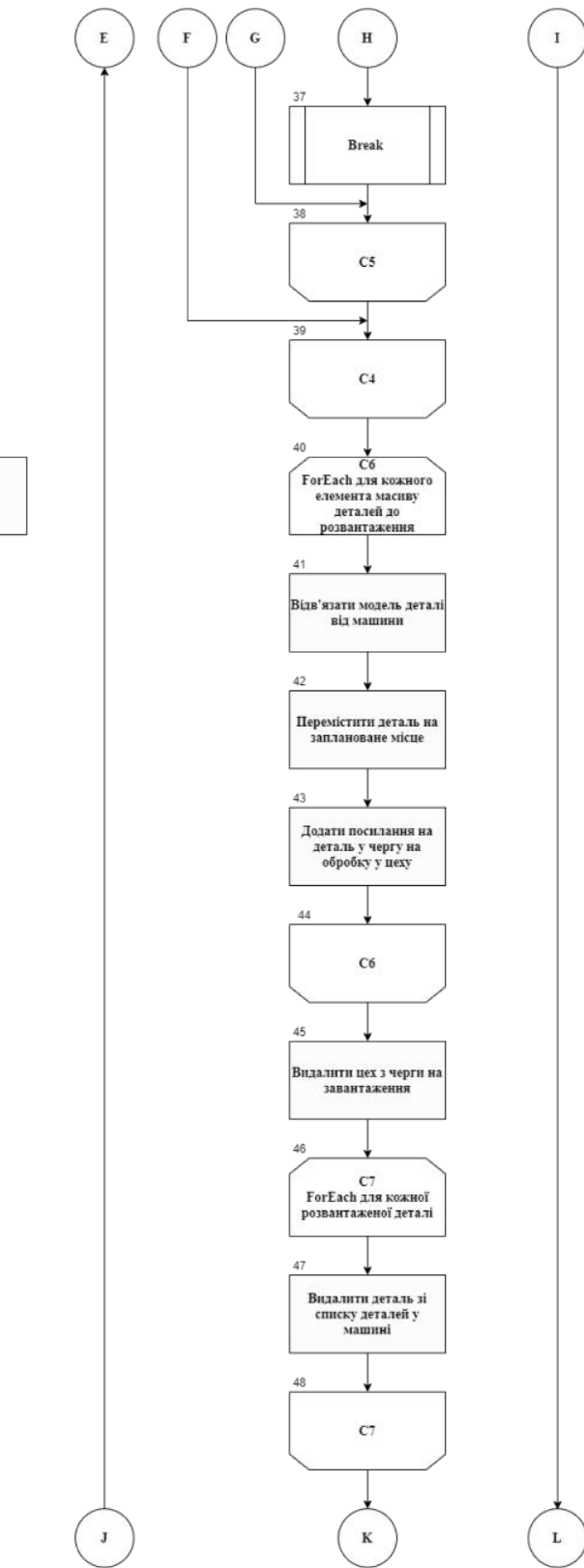
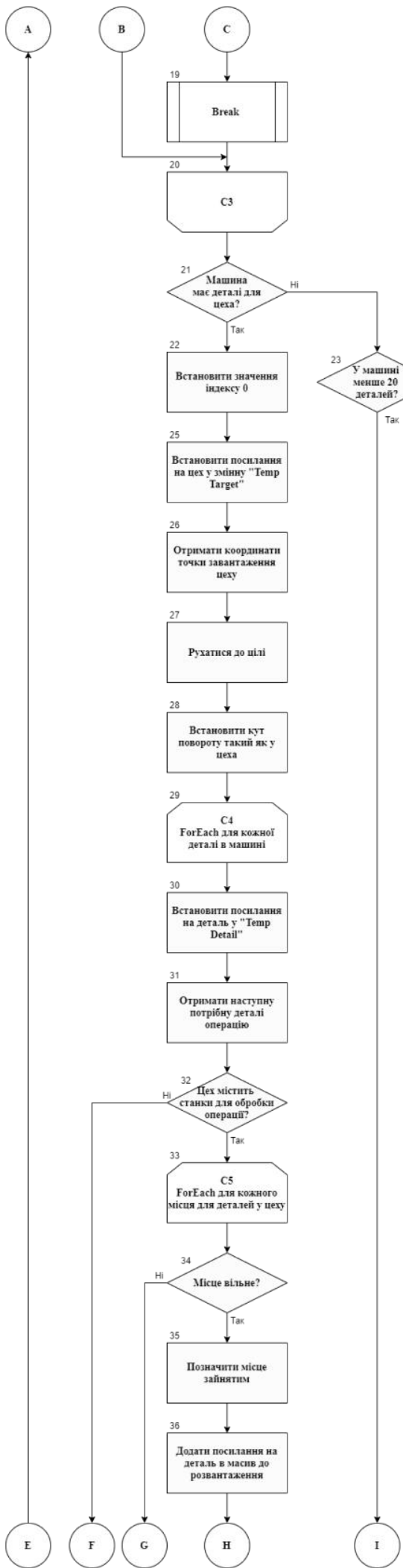
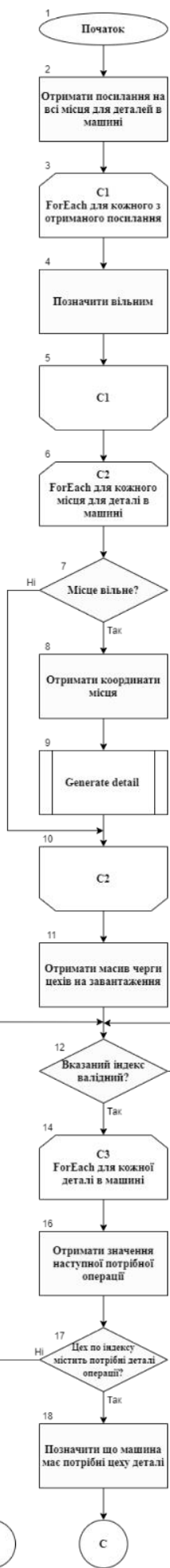
Кафедра  
Технічної кібернетики

Літ.	Маса	Мірило
Лист		Листів
Група ІК-91мп		

Зм.	Лист	№ докум	Підпис	Дата
Розроб.	Щербінський А.О.			
Перев.	Лісовиченко О.І.			
Н. контроль	Пасько В.П.			
Затв.	Пархомей І.Р.			

## ДОДАТОК В

Блок-схема алгоритму «Loading Truck»



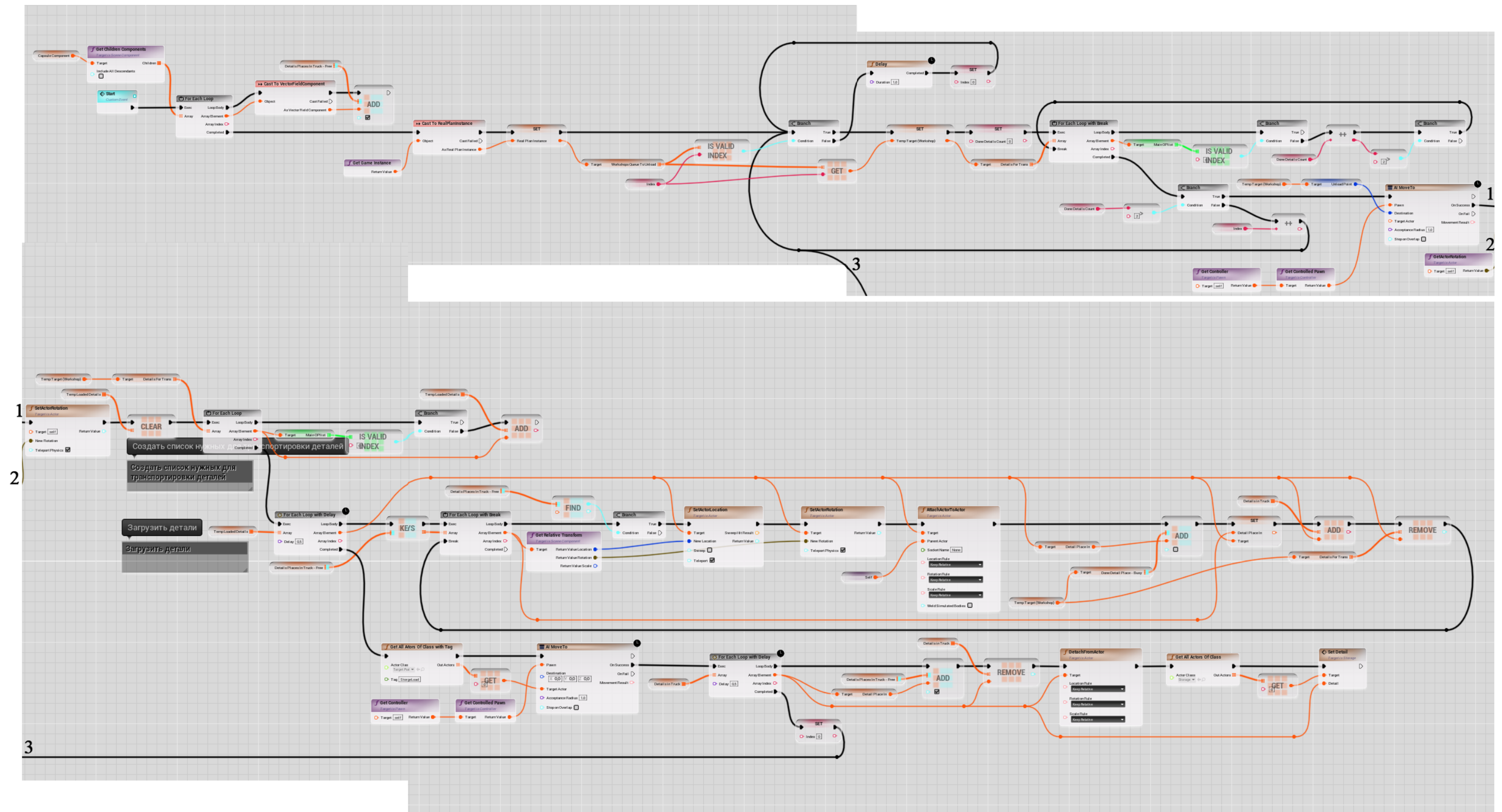
						ІК-91.32 3132.04						
						Блок-схема алгоритму «Loading Truck»			Літ.		Маса	Мірило
Зм.	Лист	№ докум	Підпис	Дата								
Розроб.	Щербінський А. О.											
Перев.	Лісовиченко О. І.											
									Лист		Листів	
						Кафедра Технічної кібернетики			Група ІК-91мп			
Н. контроль	Пасько В. П.											
Затв.	Пархомей І. Р.											



## ДОДАТОК Г

### Алгоритм «Done Details Truck»

# Алгоритм Done Details Truck



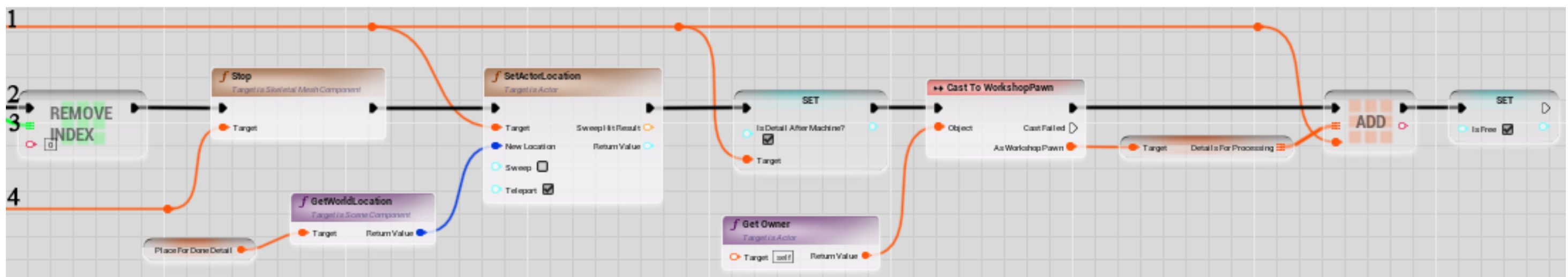
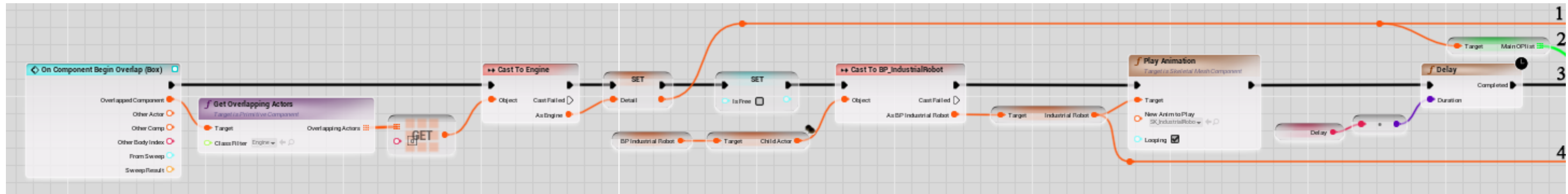
Демонстраційний плакат №1  
до магістерської дисертації на тему  
„Система моделювання гнучкого виробництва”

Розробив: Щербінський А.О.  
Прийняв: Лісовиченко О.І.

## ДОДАТОК Г

### Алгоритм «Machine»

# Алгоритм Machine



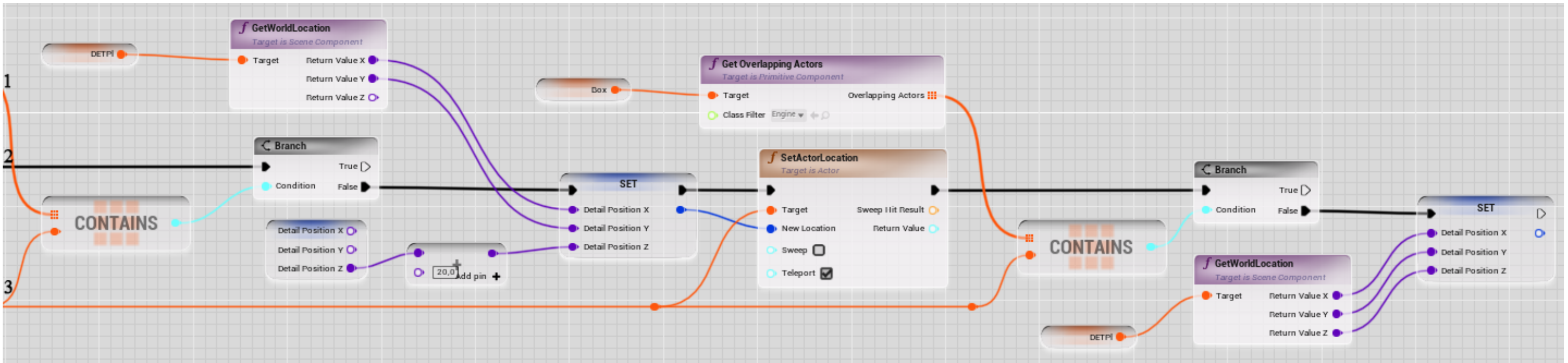
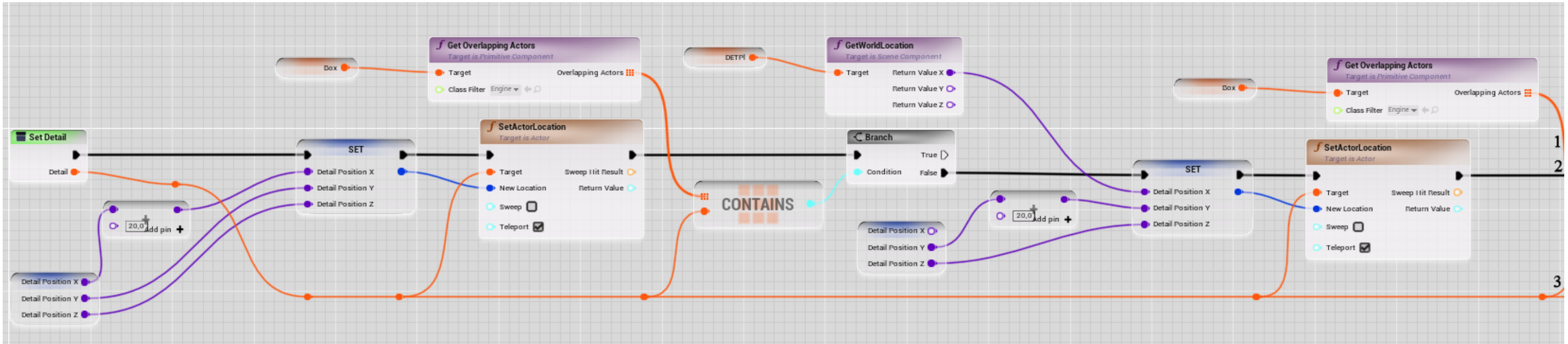
Демонстраційний плакат №2  
до магістерської дисертації на тему  
„Система моделювання гнучкого виробництва”

Розробив: Щербінський А.О.  
Прийняв: Лісовиченко О.І.

ДОДАТОК Д

Алгоритм «Storage»

# Алгоритм Storage



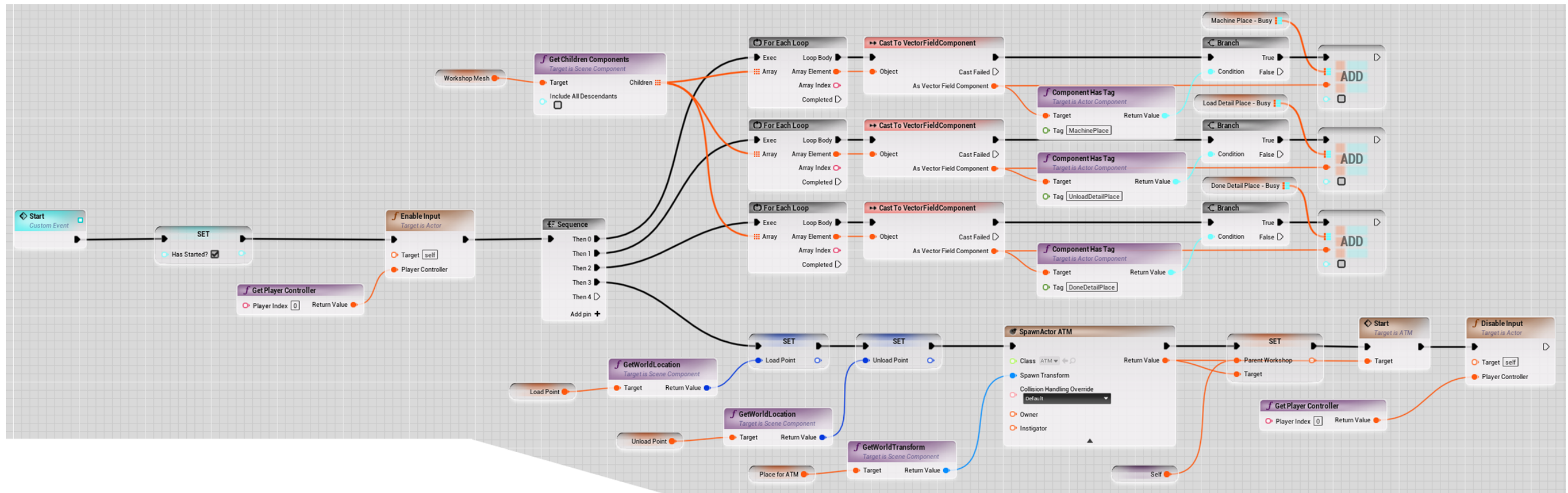
Демонстраційний плакат №3  
до магістерської дисертації на тему  
„Система моделювання гнучкого виробництва”

Розробив: Щербінський А.О.  
Прийняв: Лісовиченко О.І.

## ДОДАТОК Е

### Алгоритм початку роботи «Workshop»

# Алгоритм початку роботи Workshop



Демонстраційний плакат №4  
до магістерської дисертації на тему  
„Система моделювання гнучкого виробництва”

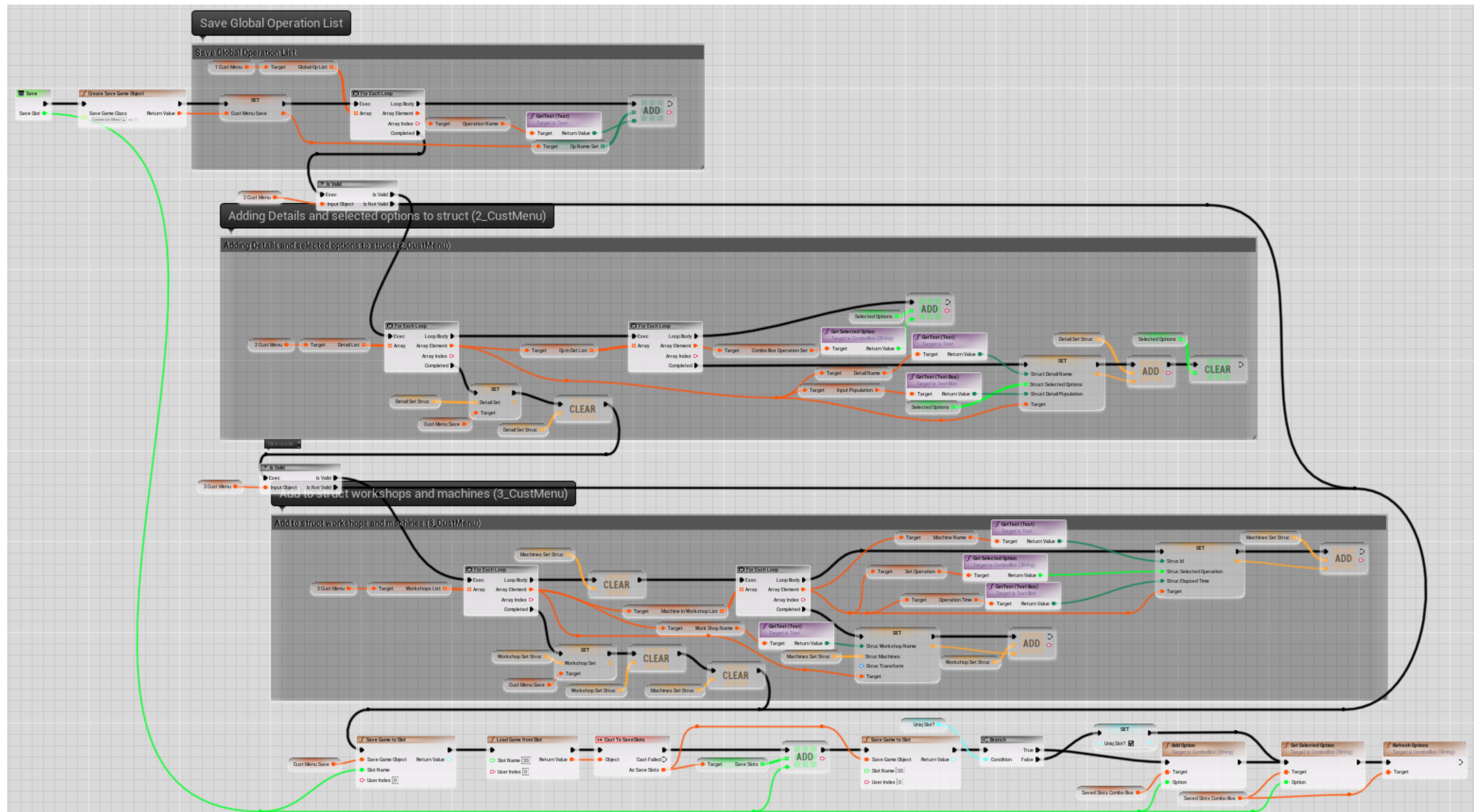
Розробив: Щербінський А.О.  
Прийняв: Лісовиченко О.І.



## ДОДАТОК Є

### Алгоритм функції «Save»

# Алгоритм функції «Save»



Демонстраційний плакат №5  
до магістерської дисертації на тему  
„Система моделювання гнучкого виробництва”

Розробив: Щербінський А.О.  
Прийняв: Лісовиченко О.І.

## ДОДАТОК Ж

Результати перевірки «Unichesk»

Ім'я користувача:  
Лісовиченко Олег Іванович

ID перевірки:  
1005352775

Дата перевірки:  
03.12.2020 18:20:01 EET

Тип перевірки:  
Doc vs Library

Дата звіту:  
03.12.2020 18:22:42 EET

ID користувача:  
76913

Назва документа: Щербінський\_03122020\_PL

Кількість сторінок: 104 Кількість слів: 15793 Кількість символів: 120361 Розмір файлу: 29.83 MB ID файлу: 1005645136

## 9.16% Схожість

Найбільша схожість: 2.36% з джерелом з Бібліотеки (ID файлу: 1000042610)

Пошук збігів з Інтернетом не проводився

9.16% Джерела з Бібліотеки

513

Сторінка 106

## 0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

## 0% Вилучень

Немає вилучених джерел